# Treball de Fi de Grau

**Grau en Enginyeria en Tecnologies Industrials (GETI)**

# Development of an Optimal Control Framework to Predict Human Motion

## MEMÒRIA

21 de juny de 2020


**Autor:** Natalia Rina García

**Directors:** Josep Maria Font Llagunes

Míriam Febrer Nafría


**Convocatòria:** Juny 2020

ETSEIB


Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona


UPC

## Abstract

Motion prediction is a field of great interest in sports and rehabilitation because it can help to determine the best way to perform a sport movement, to choose the optimal treatment for a specific patient, or to personalize an assistive device.

The main objective of this bachelor's thesis is to develop an optimal control framework to predict human motion. The biomechanical simulation software OpenSim has been used to develop the biomechanical model and perform inverse dynamics analysis, and the direct collocation optimal control software Moco has been used to formulate and solve the optimal control simulations. Once the optimal control framework is developed, it has been used to predict human motions, in this thesis, the squat-to-stand and the walking 2D motion. Their cost functions have been modified to predict the most accurate motion.

This project includes a theoretical background, a description of what is human motion, the two target motions, the analysis types and the optimal control techniques for human motion prediction. We explain the models used, the reference data and the framework of OpenSim Moco. For every motion, the biomechanical model and the problem formulation are presented in detail, with their results and discussion. Finally, we exposed the environmental, social and economic impact of the project.

Regarding the results of this thesis, the most accurate motion is reached by implementing the state tracking goal. It also has been appreciated that the kinetic energy goal improves convergence but it needs some kinematic constraints for a correct motion execution. Moreover, the control goal complemented with tracking does not bring any extra help but it could be a useful cost term for prediction without or little tracking. Regarding the reality of the motion predictions, the squat-to-stand one has been executed as a real one, whereas the gait has not been real since a contact model was missing and, besides, the model used is not a full-body one.

# Contents

## List of Figures

## List of Tables

# 1 Introduction

## 1.1 Motivation

I am in fourth grade, the last one of my degree in *Industrial Engineering*. I am studying that degree due to I love mathematics and physics, but now that I am finishing it, I have to think about what would I want to do in the future, that is Bio-medical Engineering. I want to be useful to society in a direct way: influencing people's health with my knowledge of engineering. With this desire in my heart, I received a mail from my tutors proposing to me this thesis and I confirmed immediately. Moreover, my uncle had an accident several years ago and nowadays he needs to walk with an orthopedic leg. That is the reason why Biomechanics, the field of my thesis, really appeals to me.

Biomechanics is the area of research that applies multibody system dynamics tools for studying human motion. It has a huge variety of applications, from the athletic motions to the microbiological movements of cells. This thesis entitled *Development of an Optimal Control Framework to Predict Human Motion* corresponds to a Bachelor Thesis of the degree in Industrial Engineering. Its framework is the project entitled *Design of personalized robotic and neuroprosthetic wearable systems for walking assistance using a predictive simulation framework* which reference is RTI2018-097290-B-C33. This main project is divided into 3 subprojects. This thesis is framed within the specific objective of developing predictive simulations and it corresponds to the subproject of *Universitat Politècnica de Catalunya*.

Furthermore, *CREB* is a laboratory of biomedical research well-known in the university in which I would have liked to take part in it. So I feel pretty fortunate to put a grain of sand to help the researchers with my thesis.

## 1.2 Objectives

The main objective of this bachelor's thesis is to develop an optimal control framework to predict human motion using the direct collocation optimal control software Moco.

This general objective is divided into more specific ones:

- Research optimal control, more specifically, about optimal control in human motion prediction.
- Learn how to use OpenSim Moco through a squat-to-stand prediction example.
- Develop a MATLAB code using OpenSim Moco to optimize the optimal control problem of a squat-to-stand motion based on an example.
- Develop a MATLAB code using OpenSim Moco to optimize the optimal control problem of a 2D gait walking motion.
- Modify principally the cost functions to predict the most accurate motion.
- Study the effect of using different cost function terms on the squat-to-stand and walking 2D motion.

### 1.3 State of the Art

Biomechanics is a current field of research all over the world, especially in the most known universities such as *Stanford University*, in the USA, where some researchers have been created the direct collocation optimal control software Moco [5]. Predicting human movement is a new and challenging area of research in which a lot of researchers are working on in many different ways. Many simulation methods have been studied in the past years, for example, Ezati et al. presented a thorough review of predictive simulation methods with significant emphasis on gait [24]. Moreover, the simulations can be done with data from a person wearing an assisting device such as knee prosthesis [28].

Musculoskeletal simulations are powerful tools that allow us to gain more insight on the biomechanic research, for instance, they can estimate values that are difficult to measure experimentally such as muscle tendon parameters. They can also help to isolate and test hypotheses about neural control and movement. What is useful about simulations is that they can take a variety of input data and generate tons of types of data that can be used for a wide array of research questions. It is better to model the human body in detail in order to obtain more accurate values and a perfect motion execution. However, more precision on modeling causes more time duration of the simulations. That is the reason why some articles talk about computational aspects of motion prediction such as Meyer et al. (2016) [29].

In the BIOMEC lab (CREB, UPC), before me, there were some other thesis about prediction in which I have based mine. Pallarès-López's thesis [2] is about optimal control prediction of a consistent 2D walking motion using a 2D OpenSim model (as Ackermann et al. [21]) and Febrer-Nafría's PhD thesis [1] is an amplified project including a complete 3D walking motion study. They have used the solver GPOPS-II to predict human motion. One of the following steps of the team research is to move the current methods already developed to Moco (using MATLAB language) and predict human motion. The latter is the scope of this thesis.

# 2 Theoretical Background

## 2.1 Human Motion

Human motion is very complex and it involves a rich variety of types of movement, such as the sit-to-stand [26] motion among others. In this project, two different types have been chosen: the squat-to-stand motion and a gait cycle in 2D. The squat-to-stand motion is studied even by NASA [27] and the walking 2D motion is the most studied motion in biomechanics [3].

### 2.1.1 Squat-to-stand Motion

The squat-to-stand motion is used in daily living, for instance, when picking up something from the floor. It is also used in many sports by the athletic population, especially in the gym. In this motion, the lower-body muscles work, such as the quadriceps femoris and the hip extensors. Moreover, some other muscles from the upper-body are used during the motion, such as the abdominals [30].

The squat-to-stand motion has two phases and it consists of going from one position to another and then the way back, in other words, the initial position is to squat, then the model archives the stand position and next to the final one which is to squat, see Figure 1.

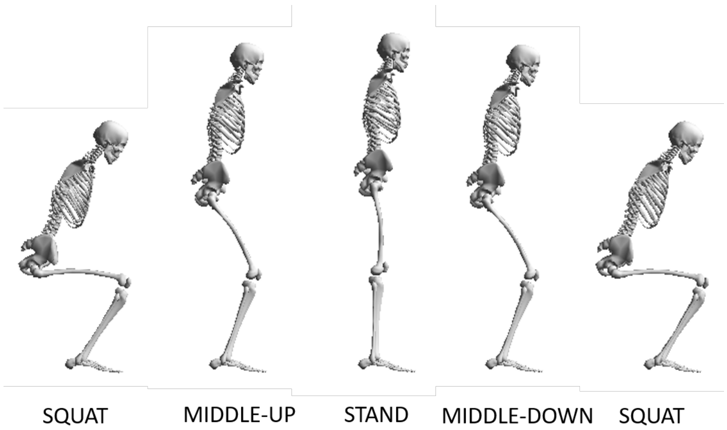



Figure 1: Frames of the squat-to-stand motion. Extracted from one of the simulations of the thesis.

### 2.1.2 Walking 2D Motion: Gait Cycle

The gait is the most common human motion by the whole population except for babies and disabled people. Walking is definitely in our daily life. People walk at home, through the streets, going from one place to another on foot...

It is more complex than the squat-to-stand motion, principally because the feet are not touching permanently the floor. The gait cycle (Figure 2) is made up of two general phases: the stance and the swing phases. The stance phase is the one in which both feet touch the ground. The swing phase is the part of the gait in which the body stands on one foot while the other is off the ground to take the next step.

In this thesis, the gait cycle is split up into 5 phases between these six positions: right toe off

(RTO), right mid swing (RMS), right heel strike (RHS), left toe off (LTO), left mid swing strike (LMS) and left heel strike (LHS).

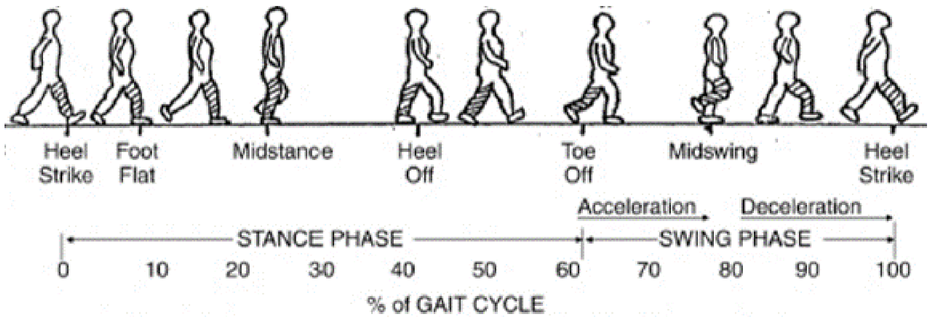



Figure 2: The gait cycle. Extracted from [31].

This motion prediction requires defining the different contact ground forces with constraints or using contact models. However, the walking simulations in this thesis do not correspond to a gait prediction because the 2D model has been simplified, thus it has the pelvis and the torso fixed to the ground. Owing to the model, the walking prediction corresponds to the legs walking while they are "flying". From the whole gait cycle, it has been predicted only the part that corresponds to an 80 % approx. Moreover, in the cost functions there is always the term of state tracking, due to the motion has been predicted only with the variable bounds and no other restriction. The experimental data used for gait tracking were collected by the research group at the Biomechanics Laboratory (CREB) of *Universitat Politècnica de Catalunya* (UPC) [1].

## 2.2 Types of Analysis

Human movement is generated starting from the neural command of the brain or spinal cord, as shown in Figure 3. The neural command is sent through excitations to the muscles and tendons which generate forces on the skeletal system. These forces generate moments about the joints and cause changes in joint angles that we observe [3].

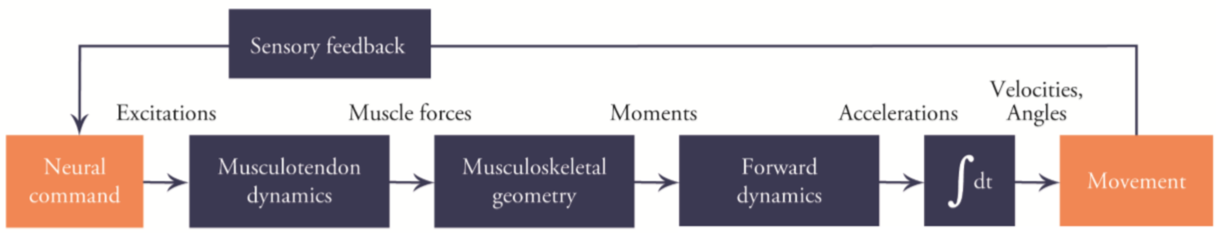



Figure 3: Human movement analysis. Extracted from [3].

As shown in Figure 3, human movement depends on precise coordination between our neural, muscular and skeletal systems. A variety of musculoskeletal tools have been developed for better understanding of the role of these systems in movements.

There are two broad classes of simulations known as forward and inverse methods. The forward analysis consists of generating movement from neutral commands; whereas inverse analysis takes data measured from observed movements to estimate the joint moments, muscle forces or neural commands that generated that movement. Forward methods can provide predictions on motions. However, they tend to take much more time to generate simulations than inverse methods.

INVERSE METHODS [32]

The inverse analysis is more commonly used. It estimates the torques, forces or neural commands needed to generate a measured motion. The input of this method is a given motion through joint angles and coordinates calculated from inverse kinematics, for example, from marker positions.

A common example of inverse methods is **inverse dynamics** whose goal is to calculate joint torques from a measured motion. It is a straightforward method and carries minimal assumptions.

FORWARD METHODS [32]

In forward methods, muscle and moments are used to compute the motion through the simulation. The calculations and optimizations are done over the whole simulation cycle rather than at each time point, for instance, the minimization of metabolic cost over a whole walking cycle which has been used often for generating simulations of walking with muscle weakness and contracture [18]. Since optimizations are done over a whole simulation, multiple passes of the simulation must be performed which comes with a high computational cost.

One of the most important is **forward dynamics** which consists of generating a motion based on specified neural commands such as muscle excitations. This input of neural commands can (but not necessarily) be from an experiment. Forward dynamics can be easy to implement for simpler motions. However, for more complex tasks like walking it is difficult to use without a controller. For forward dynamics, hand-tuning controls are more difficult than for inverse dynamics.

## 2.3 Human Motion Prediction

One of the most common methods of human motion prediction is direct collocation which aims to generate movements with a lower computational speed than inverse methods. In an easy way, it can be said that it consists of a mixture of the inverse and the forward dynamics analyses. It is a method that concurrently optimizes the whole motion trajectory and neural command. In this method, intermediate solutions do not satisfy physical constraints, although these physical constraints are satisfied by the end of the optimization. So direct collocation aims to generate a motion based on a high-level tasks quantified by an objective function.

Direct transcription methods are used by direct collocation and they are able to discretize a continuous trajectory optimization problem by approximating all of the continuous functions in the problem statement as polynomial splines [13]. The ones used by OpenSim Moco are trapezoidal and Hermite-Simpson transcriptions; for more details on them see the Theory Guide of OpenSim Moco [6].

OpenSim Moco uses direct collocation because it relies on gradient-based optimization, and therefore converges faster and more reliably when all functions in the optimal control problem are continuous and differentiable [5].

Direct collocation is a concrete way of optimal control. Thus, this field will be the key for defining the optimal control problem for predicting human motion. Optimization methods can be used to automatically generate controls for motions. These depend on optimizing a quantity called an objective function. Objective functions quantify high-level criteria that should be met during the simulation.

As Umberger et al. said, the human neuromusculoskeletal system can be modeled as a group of differential equations subject to dynamics and some controls that influence the behaviour of the human system. Optimal control helps to establish the controls by minimizing them using a performance criterion, for instance, minimizing the joint torques of the model. Optimal control also allows to do tracking which consists of determining the positions and orientations of the model bodies by minimizing errors between modeled (states in this project) and measured variables.

Optimal Control Problem

An optimal control problem consists of state equations, controls, constraints and the cost function. The states of the system are the minimal variables that define the system, containing all aspects of it, including the possibility of determining its future behavior by implementing the function f. A state vector, $\mathbf{y}(t)$, of a system with $n$ states is defined as

$$\mathbf{y}(t) = (\mathbf{y}_1(t), \mathbf{y}_2(t), ..., \mathbf{y}_n(t)).$$

In this thesis, the state is the set of generalized coordinates and velocities.
The controls are the variables that can influence the behavior of the system. The control vector, $\mathbf{x}(t)$, of a system with $m$ controls is defined as

$$\mathbf{x}(t) = (\mathbf{x}_1(t), \mathbf{x}_2(t), ...\mathbf{x}_m(t)).$$

The controls can be joint moments, muscle forces, muscle excitations or neural excitations. A control not necessary should be a biological control, for instance, in the case of a model with an assistive device, the controls can be the forces the assistive device brings to the body. In this thesis, the controls are going to be the joint torques; but instead of the joint torque value, a value from -1 to 1. When this value is multiplied by the optimal force, the joint torque value is obtained.
The state equations describe the dynamics of the model and are represented as first-order ordinary differential equations written as

$$\dot{\mathbf{y}}(t) = \mathbf{f}(\mathbf{y}(t), \mathbf{x}(t), t).$$

These state equations can be written on a more simple or complex way, depending on the research question the scientist is working on.

When the control variables are expressed as explicit functions of time, the methodology used is open loop control, and when the control variables are functions of the state variables then the control is called closed loop or feedback control. The open loop is the methodology where all the control inputs are determined at once at time 0, whereas in the close loop the control inputs are determined in a just-in-time fashion, in other words, depending on the measured state at time k, $\mathbf{y}(k)$. Open loop control can never give better performance than closed loop control because in the absence of disturbances, the two give theoretically the same performance. However, when there are disturbances, closed loop control will give better performance than open loop control because the initial state is often not known precisely and may also be random, the state equations are not known precisely and models are often only approximations of reality. Nonetheless, when a system is well-behaved and a good model for it exists, in the majority of cases in biomechanics, open loop control is a viable strategy, especially for short time horizons. In terms of computation, open loop control is typically much less demanding than closed loop control. For example, consider a system with $N_x$ distinct states and $Nu$ distinct control inputs. There are a total of $N_u^N$ different open loop strategies and $N_u(N_u^{N_x})^{N-1} = N_u^{N_x(N-1)+1}$ different closed loop strategies. Thus, there are many more closed loop strategies than open loop ones. All these concepts of optimal control theory were acquired during a bachelor semester in ETH Zürich [22]. See Figure 4 to look at how are these methodologies implemented in biomechanics.

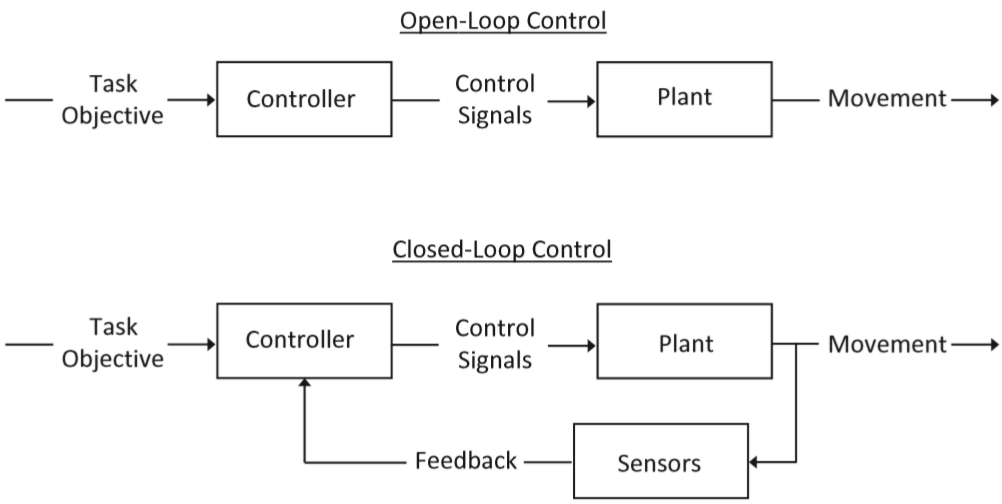



Figure 4: Flowcharts for the open loop and closed loop control of musculoskeletal movement. In open loop control, the controller generates control signals with the intent of producing a desired movement of the musculoskeletal plant. No information about the actual movement is available to the controller as the movement is happening. In closed loop control, the controller can potentially modify the control signals that are generated based on feedback from sensors about the ongoing movement. Extracted from [19].

As no sensors have been used in this thesis to obtain movement feedback, the problems will be based on open loop control.

The constraints of the optimal control problem define the size of the feasible solution domain. They can be equalities, inequalities and bound constraints. On the one hand, equality constraints, $\Phi(\mathbf{y}(t)) = 0$, make the conditions to be satisfied exactly in the final solution, for instance, the requirement of the periodicity of a cyclical motion such as walking. In this case, the values of the state at the initial time must be equal to the values of the state at the final time with some tolerance. On the other hand, inequality constraints are the same as the equality ones but with greater than or equal to, or less than or equal to, $\Upsilon(\mathbf{y}(t)) \leq 0$. They are useful to maintain the model to be real, for instance, not to fold the leg more than it biologically can. And the bound constraints are the interval where the values of the unknown variables should be. It is common to set bound constraints on the states and controls: $\mathbf{y}_{min} \leq \mathbf{y}(t) \leq \mathbf{y}_{max}$ and $\mathbf{x}_{min} \leq \mathbf{x}(t) \leq \mathbf{x}_{max}$.

To sum up, an optimal control problem is defined to find the states and controls that fulfill a specific performance criterion minimizing a determined cost function such as:

$$\mathbf{J} = \Omega(\mathbf{y}(t_f), t_f) + \int_0^{t_f} \Psi(\mathbf{y}(t), \mathbf{x}(t), t)dt$$

where $t_f$ is the final time, $\Omega$ is a scalar function evaluated at the final time, and $\Psi$ is a scalar function integrated over the entire period time. In this thesis, it is principally used the latter term.

As said before, the performance criteria are referred to a cost function which is minimized to predict human motion. One of these criteria could be tracking which consists of determining the optimal controls that make the model follow a reference trajectory, in other words, to minimize the absolute error between model generalized coordinates and velocities and their experimental counterparts. This performance criterion is easy to simulate and it does not come with computational issues. Tracking consists of minimizing the difference between the predicted motion trajectories and the reference data. That is the reason why tracking is useful to start predicting motions. Prediction determines the optimal controls so that the physiological quantities are minimized. It is really useful especially when a certain motion cannot be experimentally measured. However, finding a suitable performance criterion for motion prediction is not an easy task. In this thesis, the motions are going to be predicted with a base of state tracking.

# 3 Methodology

## 3.1 OpenSim Moco [6]

In order to solve the optimal control problem, Moco has been used. It is a part of OpenSim [4] for solving optimal control problems for musculoskeletal systems defined as OpenSim models. Its core library is written in C++. OpenSim Moco can be downloaded freely for Windows and Mac from SimTK and GitHub, where the creators develop the project and others can report bugs and request features. The documentation for Moco contains a User Guide, Theory Guide, Developer Guide, and an Application Programming Interface (API) Reference [6].

Moco is the first musculoskeletal direct collocation tool to handle kinematic constraints, which are common in musculoskeletal models. Dembia et al. [5] designed Moco to be easy to use, customizable, and extensible, thereby accelerating the use of simulations to understand human and animal movement. With OpenSim Moco, different movements have been studied such as sprinting [14], walking up inclined slopes [15], crouch gait [16] and eye movement [17]. Moco handles models with kinematic constraints, muscle activation dynamics, compliant tendons, and compliant contact, and can minimize a combination of complex costs such as marker tracking and joint reaction loads. Moco is given the OpenSim model and the motion data as inputs and it returns the motion, the controls and the parameters as outputs, through the optimization of the cost function limited with some constraints. The structure is shown in Figure 5.

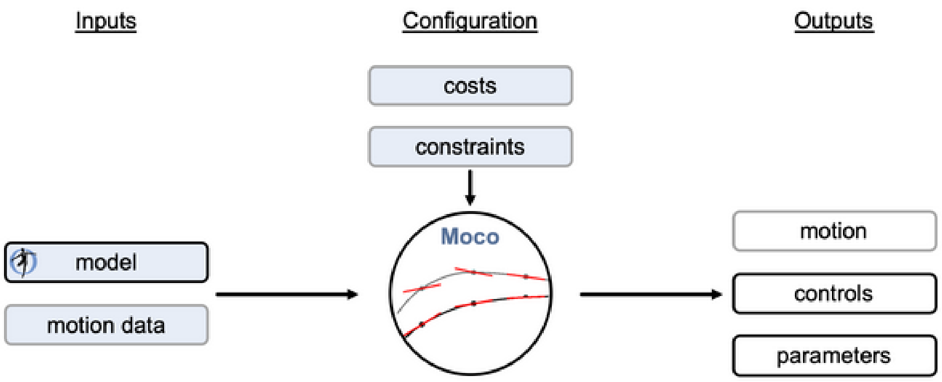



Figure 5: OpenSim Moco Structure. Extracted from [6].

Some advantages concerning other software are the following:

- Setting the model and adding a cost (termed "goal" in the example), each requires only a single statement.
- Setting bounds on states and controls by name is easier and less bug-prone than setting bounds by index, as is common in other direct collocation software.
- Bodies or muscles can be added to a model without modifying the solver, a convenience often not afforded by custom research code that couples the problem formulation to the solver.

The article *OpenSim Moco: Musculoskeletal optimal control* [5] describes how Moco implements optimal control problems for human motion prediction. The following paragraphs are extracted and adapted from this article to make the rest of the project more understandable.

Moco is a software that solves optimal control problems that can be defined using a specific library of cost and constraint modules. The general structure of OpenSim Moco consists of a study which includes the problem formulation and its solver, thus, within the same study, the optimal control problem can be formulated and solved, see Figure 6.

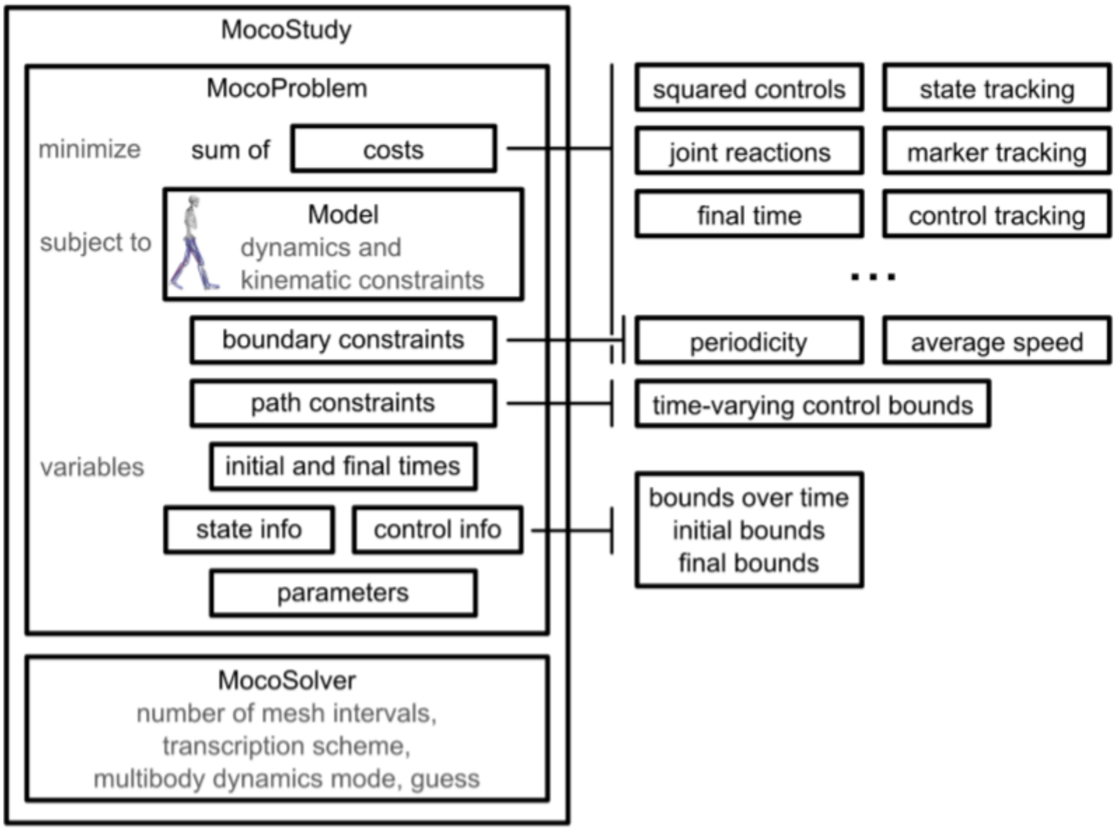



Figure 6: The structure of an optimal control study in the Moco software. Extracted from [5].

The Moco problem formulation is defined mathematically in Table 1 and it consists of the next parts:

- The cost terms: They are the term of the cost function that is minimized, for example, a sum of control effort, the differences between the states and the reference data or the joint power. Section 3.4 explains the cost terms used in this thesis.
- Multibody dynamics, muscle dynamics (e.g., mucle activation dynamics) and kinematic constraints.
- Boundary constraints: It can be fixed an average speed, the motion symmetry, or periodicity which establishes constraints between initial and final states.
- Path constraints: They are the equations of motions, for instance, forcing the residual wrench in the pelvis closer to zero [2].

- Parameter optimization: Some model parameters, such as mass or optimal fiber length, can be optimized.
- Bounds on variables: They are the bounds of the state and control values, and for the initial and final time.

| | **Equation** | **Part of the optimal problem** |
|---|---|---|
| minimize | $\sum_j w_j J_j(t_0, t_f, y_0, y_f, x_0, x_f, \lambda_0, \lambda_f, p, S_{c,j})$ | costs |
| | $S_{c,j} = \int_{t_0}^{t_f} s_{c,j}(t, y, x, \lambda, p)dt$ | |
| subject to | $\dot{q} = u$ | |
| | $M(q,p)\dot{u} + G(q,p)^T \lambda = f_{app}(t, y, x, p) - f_{inertial}(q, u, p)$ | multibody dynamics |
| | $\dot{z}_{ex}(t) = f_{\dot{z},ex}(t, y, x, \lambda, p)$ | auxiliary dynamics, explicit |
| | $0 = f_{\dot{z},im}(t, y, \dot{z}_{im}, x, \lambda, p)$ | auxiliary dynamics, implicit |
| | $0 = \phi(q, p)$ | kinematic constraints |
| | $V_{L,k} \leq V_k(t_0, t_f, y_0, y_f, x_0, x_f, \lambda_0, \lambda_f, p, S_{b,k}) \leq V_{U,k}$ | boundary constraints |
| | $S_{b,k} = \int_{t_0}^{t_f} s_{b,k}(t, y, x, \lambda, p)dt \quad k = 1, ..., K$ | |
| | $g_L \leq g(t, y, x, \lambda, p) \leq g_U$ | path constraints |
| | $y_{0,L} \leq y_0 \leq y_{0,U} \qquad y_{f,L} \leq y_f \leq y_{f,U}$ | initial and final states |
| | $x_{0,L} \leq x_0 \leq x_{0,U} \qquad x_{f,L} \leq x_f \leq x_{f,U}$ | initial and final controls |
| with respect to | $t_0 \in [t_{0,L}, t_{0,U}]$ | initial time |
| | $t_f \in [t_{f,L}, t_{f,U}]$ | final time |
| | $y(t) = (q(t), u(t), z(t)) \in [y_L, y_U]$ | states |
| | $x(t) \in [x_L, x_U]$ | controls |
| | $\lambda(t)$ | Lagrange multipliers |
| | $p \in [p_L, p_U]$ | time-invariant parameters |

Table 1: The optimal control problem that Moco can solve. Extracted from [6].

Table 1 describes the optimization problem that Moco can solve. The cost equation to be minimized consists of a summation of cost terms $J_j$ multiplied by their weights $w_j$. These cost terms depend on the initial $t_0$ and final time $t_f$, the initial $y_0$ and final state values $y_f$, the initial $\lambda_0$ and final Lagrange multipliers $\lambda_f$ and the time-invariant parameters $p$. The states include the generalized coordinates $q(t)$, the velocities $u(t)$ and the auxiliary states $z(t)$. Moreover, there are integrals $S_{c,j}$ (integrated over time, from $t_0$ to $t_f$) as addends of this summation and they depend on the time $t$, the time-dependent states $y$ and controls $x$, the Lagrange multipliers and the time-invariant parameters $p$. This cost function is subject to the multibody dynamics involving the mass matrix $M$, the applied forces $f_{app}$ and the inertial forces $f_{inertial}$ which correspond to inertial terms depending on position $q(t)$ and velocity $u(t)$. It is also subject to auxiliary dynamics expressed with an explicit ($f_{\dot{z},ex}$) or implicit ($f_{\dot{z},im}$) differential equations. There are also established some kinematic constraints with function $\phi$. Furthermore, the variables $k$ must obey boundary constraints $V_k$ where the subindex $L$ indicates the low bound and the subindex $U$ indicates the upper bound and also path constraints $g$ over the motion. In addition, the initial $y_0$ and final $y_f$ states, and the initial $x_0$ and final $x_f$ controls are bounded.

In the solver part of the study, there are implemented all the details about the solving process. Thus the problem formulation is independent of the solver that will be used. The problem formulation must require the fact that it models a multibody system. Moreover, the model can be modified without changing any aspect of the solver, that is an advantage which is not implemented in other problem solvers. Moco solver uses the CasADi library [10] to transform the optimal control problem into a nonlinear problem which is solved by IPOPT [11] and SNOPT [12]. Moco provides some functions to have easy access to the state and control values, the optimization parameters, whether the solver finds an optimal solution, the cost function value and the number of iterations of the simulation among other useful values. As seen in Figure 7, the solution of one study can be used as the initial guess of another study.

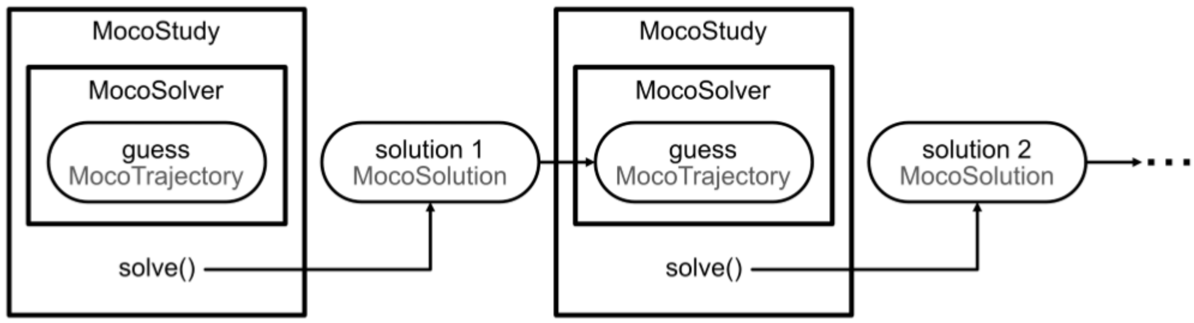



Figure 7: Problems are solved iteratively using trajectories. Extracted from [5].

Moreover, after obtaining the solution, the motion execution is visualized and the trajectories of the states and the actuators can be plot which are mostly implemented with a single line of code. The solution is written in a file with a ".sto" extension.

**Tools for standard problems** Moco provides two tools (see Figure 8) for standard problems, even though none of these tools are used in this project:

- The inverse tool that determines the controls (muscle activation or actuators) which minimize the function cost to execute exactly the prescribed motion.
- The tracking tool that determines the controls (muscle activation or actuators) which minimize the errors between the predicted and the observed motion.

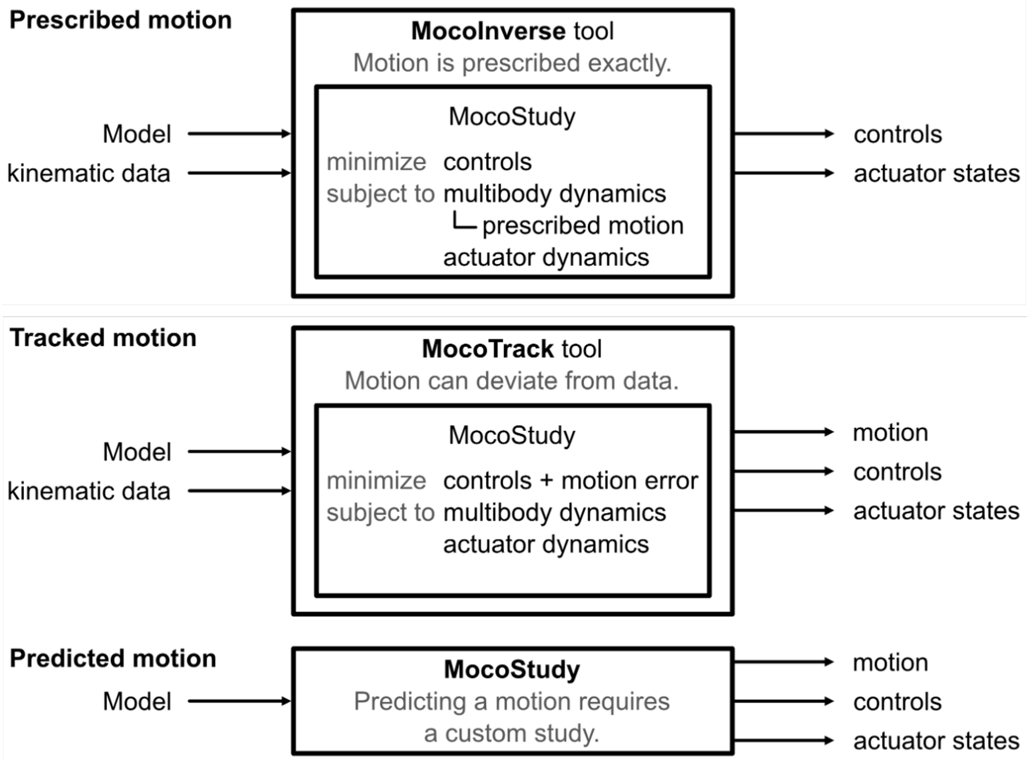



Figure 8: Tools for standard problems: MocoInverse (prescribed motion), MocoTrack (tracked motion) and MocoStudy (predicted motion). Extracted from [5].

The tracking tool is useful for predicting motions with a slight deviation from motion generated by experimental data while using the inverse tool the motion is prescribed exactly as it is observed. For tracking contact models are used, whereas measured external forces should be applied to the model in the inverse tool. The inputs for both tools are the model and the kinematic data and the outputs are the control and the actuator values, in addition to the tracking tool, the motions are also outputs of the problem. The problems that are not standard, such as motion prediction, requires a custom study which can be implemented with the Moco software in a flexible way.

In the following sections, the OpenSim model is being described adapted from [8], for better understanding of Sections 4.1 and 5.1. Then, in sections 4 and 5 the models will be described in detail, thus the formulation will be well and exactly determined.

## 3.2 Biomechanical Model

The human body is modeled as a multibody system with rigid bodies, joints and generalized coordinates. It is modeled as an OpenSim model that has its specific nomenclature. The motion study in this project is done on a skeletal level, considering the joint torques but not the muscle forces. Although only bodies, joints and generalized coordinates are explained, there are more OpenSim components that are used but not described because they are out the scope of this thesis, such as geometry, muscles, functions, frames, etc. All the missing details in these sections can be found in the OpenSim documentation [8].

### 3.2.1 Bodies

An OpenSim body is a reference frame with associated inertia specified by its mass, center-of-mass located in the reference frame, and its inertia tensor about the center-of-mass. Each body has a geometry and set of parameters that have to be specified to each body of the OpenSim model.

### 3.2.2 Joints

An OpenSim joint is a component of the model which connects two bodies (or frames) together and specifies their relative permissible motion as described in internal coordinates.

The relative motion (including the values of coordinates) is defined by specific joints, which specify the permissible kinematics of a child joint frame (on a child body) with respect to a parent joint frame (on a parent body). The designation of parent and child are used only to identify the directionality of the joint and in which frame the joint coordinates are expressed. Each joint has its relative permissible motions. There are 6 in total: 3 rotations around the x-, y- and z-axis, and 3 translations through the x-, y- and z-axis. These relative motions are referred to the transformation of the child body with respect to a parent body, thus the axes correspond to the parent body.

There are several types of joints. The ones used in this thesis are described next:

The **pin joint** provides one degree of freedom about the common z-axis of the joint frames in the parent and child bodies. If it is required a rotation about a different direction, it is necessary to rotate the joint and body frames such that the z axes are in the desired direction.

The **custom joint** a generic joint representation, which can be used to model both conventional (pins, slider, universal, etc.) as well as more complex biomechanical joints. The behavior of a custom joint is specified by its spatial transform that can be either one of the 3 rotations or one of the 3 translations that define the spatial position of the child frame with respect to the parent frame as a function of coordinates. Each transform axis has a function of joint coordinates that describes the motion about or along the transform axis.

The **weld joint** has no relative motion of bodies. They are often used to create composite bodies from smaller simpler bodies. That is the child body will be fixed to the pared body.

For more information about classes and OpenSim nomenclature see reference [8], from which the majority of the information above has been extracted.

### 3.2.3 Generalized Coordinates

The generalized coordinates are the independent variables that uniquely describe the model configuration with respect to a reference. When the number of generalized coordinates is equal to the degrees of freedom, the system is holonomic which is the case of the models of this project. It is common to express the generalized coordinates, their velocities and accelerations as $q$, $\dot{q}$ and

$\ddot{q}$. In this project, the state corresponds to the set of generalized coordinates and velocities. The accelerations are not used as variables of the optimal problem in this thesis.

### 3.3 Experimental Data

Squat-to-stand data

The experimental data used for tracking are different for the two motions studied. The data of the squat-to-stand motion have been generated with some sinusoidal functions defined as:

$$f(t) = \alpha_0 + Asin(wt) \tag{1}$$

where $\alpha_0$ is the initial value of each state, $A$ is the amplitude of the state trajectory, $t$ is a vector of 100 components from 0 to 2 seconds, and $w = \frac{2\pi}{T} = \frac{2\pi}{4}$ as $T$ is the period of the sinusoidal function, the double of the final time value. See the values for each parameter in Table 2.

| Function parameters | Right hip flexion | Right knee angle | Right patellofemoral angle | Right ankle angle |
|---|---|---|---|---|
| $\alpha_0$ [rad] | -2 | -2 | 2 | -0.5 |
| $A$ [rad] | 1.8 | 2 | -2 | 0.5 |

Table 2: Parameters of Equation 1 to obtain data for the generalized coordinates of the squat-to-stand motion.

Walking 2D data

The experimental data used for the walking 2D motion is the data that a member of the research team had obtained in 2018 [1]. The experimental data were collected at the Biomechanics Laboratory (CREB) of *Universitat Politècnica de Catalunya* (UPC). The lab equipment consists of a motion capture equipment OptiTrakTM from NaturalPoint Inc, designed to capture the position of points in a 3D space. The system disposes of 16 cameras V100:R2 model, which incorporate infrared light LED's, and markers which are little spheres covered by reflective material. The procedure consists of a person with markers attached to his/her body walking through the lab. The cameras are located all over the lab: eight at 3 meters from the floor and the rest at a meter and a half; this layout guarantees a precise capture of gait. The cameras emit light, the markers reflect the light and the optical system of the camera receives this reflected light in a sampling rate of 100 Hz. With the information of all the cameras, the position of all markers is obtained in a 3D space. The lab equipment also has two force plates AMTI Accugait which can determine the contact wrench applied to the person who walks on them. For more details on the laboratory equipment, motion capture and data processing, the reader is referred to [2].

After data processing, the experimental data consist of a set of vectors: a vector of values for each generalized coordinate, another of their velocities, another of their accelerations and another one for the jerks (the third derivative of the generalized coordinates). There are also vectors of the residual forces and moments (the residual wrench components), of the joint torques and of the ground reaction forces. Each vector is made of 109 components and it corresponds to a motion from 0.67 to 1.75 seconds of walking 2D.

### 3.4 Moco Cost Terms for the Cost Function [5]

The optimal control problem consists of minimizing a cost function split into cost terms, also named goals. Some are already in the Moco library, if not, the costs can be implemented by a C++ plugin following the same procedure as for OpenSim plugins. Moco permits to combine cost terms to define the cost function. In this thesis, only cost terms of the providing library have been implemented; they are described in this section.

CONTROL GOAL

The control goal consists of minimizing the sum of the absolute value of the controls raised to a given exponent, integrated over the phase.

$$\frac{1}{d}\int_{t_i}^{t_f}\sum_{c} w_c|x_c(t)|^p dt$$

where $x_c(t)$ is the control signal; the weight $w_c$ of each control $c$ can be determined, if not the default one is 1; the exponent $p$ must be an integer greater than or equal to 2, and is 2 by default. If the cost term is used for a predictive simulation, it should be divided by displacement (there is an already implemented function for this); otherwise, this cost is minimized by not moving. Dividing by displacement leads to a quantity similar to the cost of transport [5].

STATE TRACKING GOAL

The state tracking goal consists of minimizing the squared difference between a state variable value and a reference state variable value, summed over the state variables for which a reference is provided, and integrated over the phase. The reference should be data provided as a file name (e.g., with a ".sto" extension) or as a TimeSeriesTable (a type of table provided by OpenSim). These data should be filtered because tracking problems in direct collocation perform better when tracking smooth data; the data of the walking 2D motion taken from [1] was already filtered.

OUTPUT GOAL, SPECIFICALLY KINETIC ENERGY GOAL

The output goal uses any output of the model as the integrand of a goal to be minimized. In this thesis, the output to be minimized is the kinetic energy.

# 4 Squat-to-stand Prediction

In this Chapter, the OpenSim model and the problem formulation to predict squat-to-stand motion are presented. Finally, the results are exposed and discussed.

As mentioned in Section 2.1.1, the squat-to-stand motion (see Figure 9) consists of a motion which its initial position is to be squatting, the medium position is to be upright standing and its final position is to squat again. Rigorously, the motion should be named squat-to-stand-stand-to-squat motion, but in this thesis, it will simply be called squat-to-stand motion.

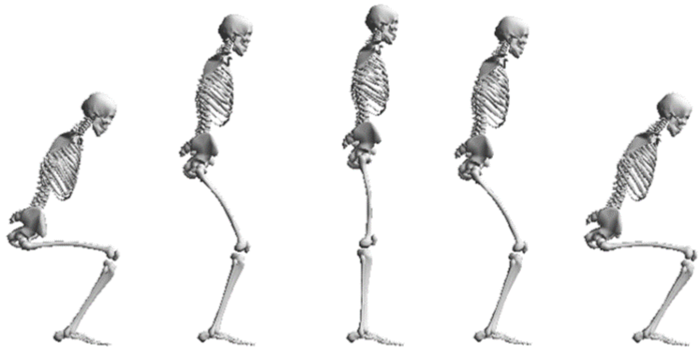



Figure 9: Events of the squat-to-stand motion. Extracted from one of the simulations of the thesis.

## 4.1 Model

For the squat-to-stand prediction, the $squatToStand\_3dof9musc.osim$ model is used [33]. It has 8 bodies, 8 joints, 4 generalized coordinates and 3 degrees of freedom. The base model is the Rajagopal 2016 [7] model reduced to 3 DOFs which contained a torso and a right leg with 9 muscles obeying activation dynamics and compliant tendon dynamics, using the maximum isometric forces from Carmichael Ong's *gait8dof18musc.osim* model. To enforce mediolateral symmetry, a single leg and doubled muscle strengths are used, which are obtained from [20]. Even though the base squat-to-stand model has 9 muscles, the model used in this thesis has the muscles removed. The squat-to-stand motion is studied from a skeletal level, without taking into account the muscle forces. That is the reason why the muscles are not in the scope of the project.

The squat-to-stand model is composed of the ground and 8 rigid bodies: the pelvis, the right femur, the right tibia, the right patella, the right talus, the right calcn, the right toes and the torso. The model has 8 joints. In Table 3, the joints of the model are list with their name in OpenSim, the degrees of freedom (DOF) the joint has, the type of joint, the child and the parent frame [1]. The generalized coordinates of this model are the joint movements named in OpenSim as $ankle\_angle\_r$, $knee\_angle\_r$, $hip\_flexion\_r$ and $knee\_angle\_r\_beta$. The $ankle\_angle\_r$ is the angle the ankle has, that is the angle that has the tibia with respect to the talus. The $knee\_angle\_r$

[1]This nomenclature is used to express that child bodies move with respect to parent bodies

is the angle the knee has, that is the angle that has the femur with respect to the tibia. Finally, the $hip\_flexion\_r$ is the angle the hip has, that is the angle that has the pelvis with respect to the femur. Finally, the $knee\_angle\_r\_beta$ is the angle the patellofemoral has, that is the angle that has the patella with respect to the femur.

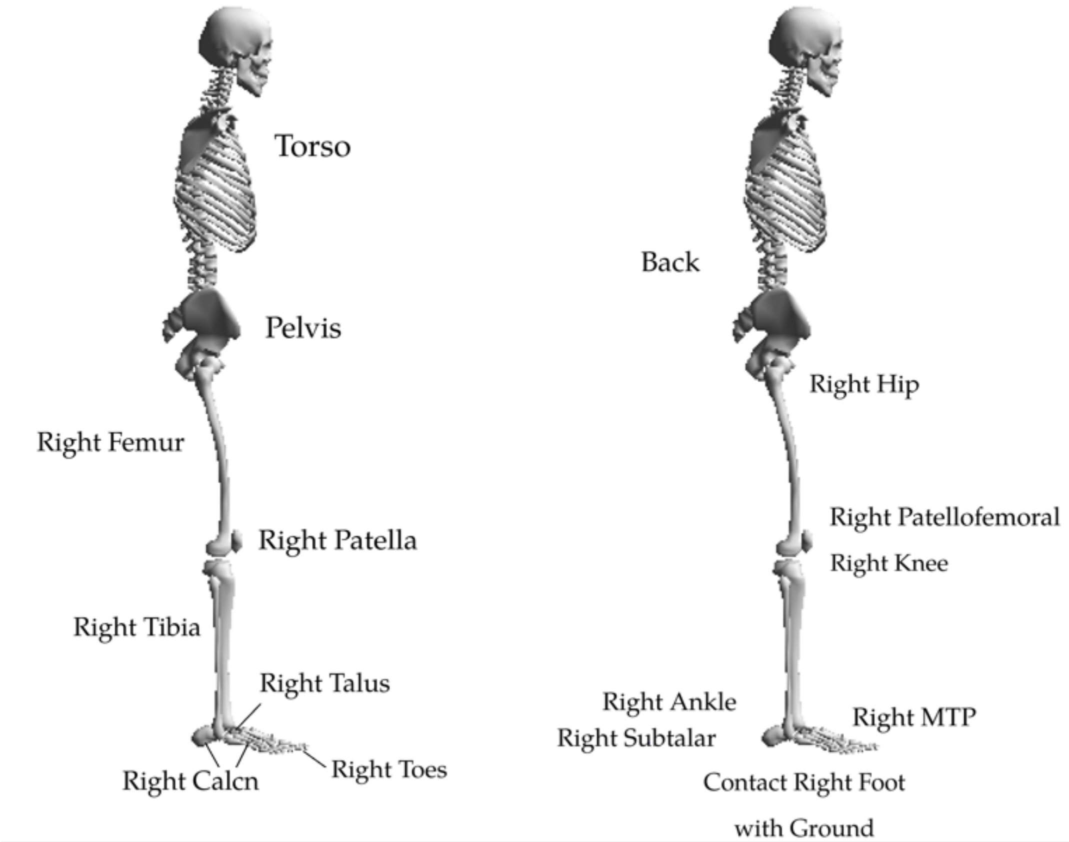



Figure 10: Illustration of the bodies (left) and the joints (right) of the squat-to-stand model. Adapted from one of the simulations of the thesis.

| Joint | Name of the joint in OpenSim | DOF | Type of the joint | Child Frame | Parent Frame |
|---|---|---|---|---|---|
| Contact Right Foot with Ground | $foot\_ground\_r$ | 0 | Weld Joint | Right Calcn | Ground |
| Right Knee | $knee\_r$ | 1 | Custom Joint | Right Femur | Right Tibia |
| Right Patellofemoral | $patellofemoral\_r$ | 1 | Custom Joint | Right Patella | Right Femur |
| Right Ankle | $ankle\_r$ | 1 | Pin Joint | Right Tibia | Right Talus |
| Back | $back$ | 0 | Weld Joint | Torso | Pelvis |
| Right Subtalar | $subtalar\_r$ | 0 | Weld Joint | Right Talus | Right Calcn |
| Right MTP* | $mtp\_r$ | 0 | Weld Joint | Right Toes | Right Calcn |
| Right Hip | $hip\_r$ | 1 | Pin Joint | Pelvis | Right Femur |

Table 3: Joints of the model, their name in OpenSim, the DOFs of the Child Frame with respect to the Father Frame, and the type of the joint. *MTP means the metatarsophalangeal joints [9].

Although Table 3 shows that the model has 4 degrees of freedom, that is not certain at all because one degree of freedom is restricted. There is a relation between two generalized coordinates: the right patellofemoral angle and the right knee angle. Thus, the whole configuration of the model has 3 degrees of freedom. This relation is a constraint between the $knee\_angle\_r$ (the independent coordinate) and the $knee\_angle\_r\_beta$ (the dependent one), thus only the $knee\_angle\_r$ coordinate brings a DOF to the model. First, the right knee and the right patellofemoral has one DOF each (see Table 3). But with this the kinematic constraint, these two DOFs become to be only one. Consequently, the whole model has only 3 DOFs.

## 4.2 Problem Formulation

The squat-to-stand problem formulation is based on the "torque-driven predictive problem" of the example of the OpenSim Moco documentation called $exampleSquatToStand.m$ [33]. In this section, the states and controls are presented. Moreover, the different cost functions and constraints are defined.

There are 8 states: 4 generalized coordinates and 4 velocities of the following angles: the $knee\_angle\_r$, the $ankle\_angle\_r$, the $knee\_angle\_r\_beta$ (which corresponds to the patellofemoral joint) and the $hip\_flexion\_r$.

The controls of the problem are the three actuators of Table 4: one in the hip, another in the knee and the last one in the ankle. Table 4 shows the name in Opensim of the actuators, their corresponding coordinate name in OpenSim of the model and the optimal force that is established to the actuator. The minimum and maximum control value of all the actuators of Table 4 is set to -1 and +1. In summary, three actuators are added which corresponds to the 3 DOFs of the model.

| Actuator | Matching coordinate | Optimal Force [in N] |
|---|---|---|
| $tau\_hip\_flexion\_r$ | $hip\_flexion\_r$ | 150 |
| $tau\_knee\_angle\_r$ | $knee\_angle\_r$ | 300 |
| $tau\_ankle\_angle\_r$ | $ankle\_angle\_r$ | 150 |

Table 4: Description of the actuators, with their matching coordinates and their optimal force.

The initial and final time bounds are 0 and 2 seconds. Consequently, at time equal to 1 second, the model should be upright standing.

The cost terms (explained in Section 3.4) used for squat-to-stand prediction are shown in Table 5. The control term minimize the effort of the controls which are the three joint torques of the hip, the knee and the ankle. The kinetic energy term minimize the kinetic energy that the model requires to execute the motion. And the state tracking term minimize the squared difference between the state variables and the data generated with the sinusoidal functions, see Section 3.3. Six different cost functions (Table 5) are applied and studied in the squat-to-stand problem: the single term ones (control goal, kinetic energy (KE) goal and state tracking goal), the two terms ones (control and state tracking, kinetic energy and state tracking) and the three terms one (control, kinetic energy and state tracking).

| Cost function | | | |
|---|---|---|---|
| Type | Control | KE | StateTrack |
| Single term | X | | |
| | | X | |
| | | | X |
| Two terms | X | | X |
| | | X | X |
| Three terms | X | X | X |

Table 5: Table of the different cost functions studied in the squat-to-stand motion.

The constraints of the problem are the variable bounds. The position bounds of the generalized coordinates are set to the problem and they are shown in Table 6. Initial and final bounds are the same for each coordinate due to the model starts in the same position as it ends: in the squat position. The trajectory bounds of all the velocities are set to the default ones ([-50, 50] for all the states, except for the value of the coordinate $knee\_angle\_r\_beta$ of the joint $patellofemoral\_r$ which bounds are [-5, 5] and it has not initial and final value). Their initial and final bounds are forced to be 0, due to all model coordinates should start and end at rest.

| Name of the general coordinate in OpenSim | Trajectory bounds [deg] | Initial bound [deg] | Final bound [deg] |
|---|---|---|---|
| *hip_flexion_r* | [-114.6, 28.7] | -114.6 | -114.6 |
| *knee_angle_r* | [-114.6, 0] | -114.6 | -114.6 |
| *ankle_angle_r* | [-28.7, 40.1] | -28.7 | -28.7 |

Table 6: Position bounds of the generalized coordinates of the squat-to-stand problem.

The initial guess used is the Moco's default one in which each variable's value is the midpoint of the variable's bounds, the number of mesh intervals is 25 and, convergence and constraint tolerances are set to $10^{-4}$.

### 4.3 Results and discussion

The most interesting results of the simulations for the squat-to-stand motion are shown in Table 7 which allows comparisons between the six different cost functions in terms of convergence, motion effort (reflected as the cost function values) and the motion execution (reflected as the ranges of motion of the most important generalized coordinates). Moreover, states and controls for each cost function are shown in Figures 11 and 12 (single term cost functions), and 13 and 14 (two and three terms cost functions). Finally, the results are discussed and it is stated if the motion is performed correctly in different cases. The squat-to-stand motion is performed correctly when the model from the squat position reaches the stand stance and it goes down again in a symmetric way.

| Cost function | | | | Convergence | | Cost function values | | | | ROM [deg] | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Type | Control | KE | StateTrack | Num. iter. | Time [s] | Cost function | Control | KE | StateTrack | Hip | Knee | Ankle |
| Single term | X | | | 45 | 56 | 0.45 | 0.45 | - | - | 30.53 | 114.47 | 105.19 |
| | | X | | 28 | 22 | $4.92 \cdot 10^{-4}$ | - | $4.92 \cdot 10^{-9}$ | - | $1.83 \cdot 10^{-4}$ | $9.45 \cdot 10^{-4}$ | $8.79 \cdot 10^{-4}$ |
| | | | X | 49 | 38 | 0.01 | - | - | 0.01 | 28.64 | 114.56 | 103.14 |
| Two terms | X | | X | 28 | 30 | 0.51 | 0.48 | - | 0.03 | 30.73 | 113.85 | 103.64 |
| | | X | X | 22 | 51 | 7.47 | - | 3.29 | 4.18 | 32.88 | 65.49 | 16.54 |
| Three terms | X | X | X | 19 | 46 | 8.40 | 0.86 | 3.13 | 4.41 | 25.81 | 57.81 | 20.13 |

Table 7: Table of different aspects of the results for the different cost functions studied in the squat-to-stand motion. In the column of the cost function values, the "cost function column" is the value of the whole cost function, and in the other columns, it is split up into the cost term values.

In terms of convergence, the number of iterations for the cost functions which include kinetic energy term is lower than when it is not included and for kinetic energy single term the duration of simulations is the lowest (22 s). The largest numbers of iterations correspond to control and state tracking single term goals, although when they are combined as a two terms cost function, the number of iterations is considerably reduced (from 45-49 to 28). The number of iterations is not necessarily proportional to the duration of the simulation, e.g., for the two terms kinetic energy and state tracking, the time is large (51 s) and the number of iterations is low (22).

In terms of the cost function, when the kinetic energy goal is implemented as a single term the effort is by far the lowest, indeed, it can be said that is almost zero. Whereas, in the combined cost functions, it gains a bit of effort and it takes part in the optimization problem. To state tracking goal it occurs something similar, even though the single term of state tracking is pretty greater than the single term of the kinetic energy. In contrast, the control goal effort stays approximately the same value in the case of single and two terms functions but in the three terms cost function increases.

Regarding the ranges of motion (ROM) of the generalized coordinates is a great parameter to determine if the motion is performed correctly. Table 7 shows that the ranges of motion are the same in the cost functions in which the kinetic energy cost term does not appear. In contrast, in the single term kinetic energy cost function, the ROMs are approximately zero ($\approx 10^{-4}$). Moreover, when it appears with the state tracking goal, the hip angle is maintained (28.64 and 32.88 deg), the knee angle decreases (from 114.56 to 65.49 deg), but the ankle angle decreases a lot (from 103.14 to 16.54 deg). Furthermore, looking at the cost function of three terms, meanwhile the hip angle stays almost the same (28.64 and 25.81 deg) with respect to the state tracking single cost term, the knee angle decreases (from 114.56 to 57.81 deg) and the ankle angle decreases a lot more (from 103.14 to 20.13 deg).

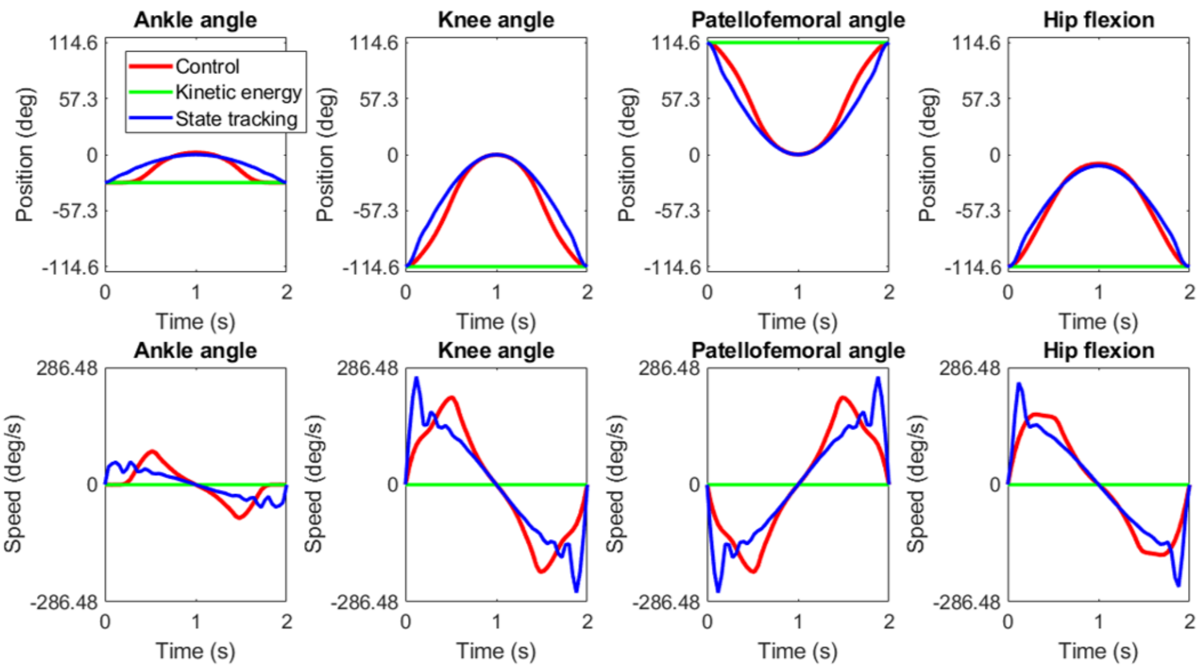



Figure 11: State variables: generalized coordinates and velocities for every different single term cost function.

Comparing the single term cost functions, in Figure 11 it is appreciated that with the kinetic energy goal, generalized coordinates are almost constant as well as seen in Figure 12 on the joint torques. With the control (in red) or the state tracking goal (in blue), the trajectories of the state values in Figure 11 are almost the same: they start and end at the same point and they have the same maximum, even though they slightly vary in the middle-up and middle-down positions. About the velocities of the control and state tracking cost functions, they look alike, even though some variations between them. The velocities using a control cost function show a maximum around the middle-up (0.5 s approx.) and a minimum around the middle-down (1.5 s approx.), whereas the velocities taken from state tracking have the maximum much earlier and the minimum much later.

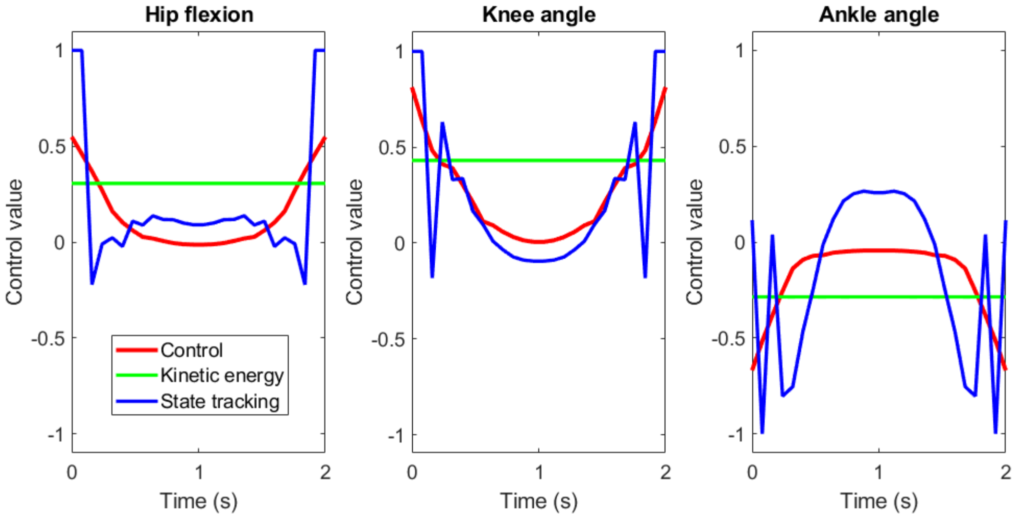



Figure 12: Actuators for every different single term cost function. Remember that the y-axis correspond to the control values (from -1 to 1). When this control value is multiplied by the optimal force, the result is the joint torque value.

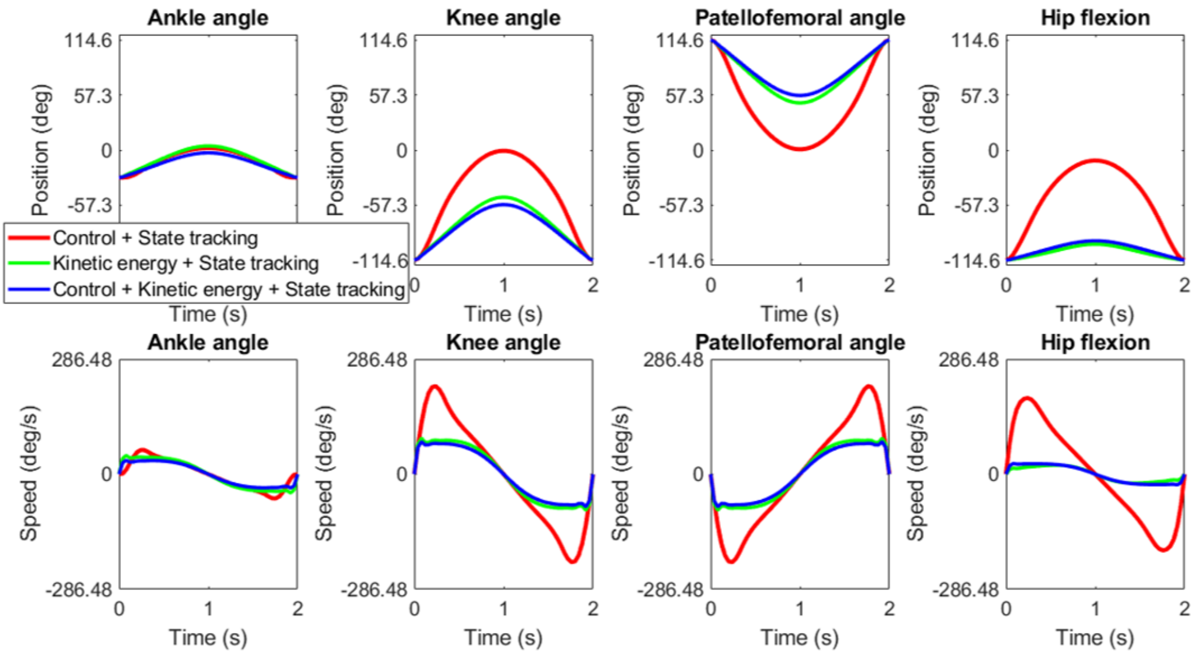



Figure 13: State variables: generalized coordinates and velocities for every different cost function with two terms.

Comparing the cost functions with two or three terms, Figure 13 represents the state values of control and state tracking (in red), of kinetic energy and state tracking (in blue) and the three cost function (in green). Notice that in the latter the ranges of motion of the state values are lower than in the red trajectories (control and state tracking), despite the angle of the ankle that has almost the same trajectories for both combinations. The cost function of kinetic energy and state tracking has less variation on the velocities than the combination of control and state tracking which presents significant maximums and minimums around the 0.25 and the 1.75 seconds, which correspond to the beginning of the upward and downward motion, despite the velocity of the coordinate of the ankle. In Figure 14 it is appreciated that the joint torques of the hip flexion and the ankle differ in the stat and end value in terms of the different cost function used, the greater difference occurs in the ankle actuator and it is about 0.5 (of control value). In the combination of kinetic energy and state tracking it is shown that the joint torques are lower in the middle point (1 s) than the other combinations, but higher in the ankle torque. Results for the three terms cost function (control, kinetic energy and state tracking) show that the ranges of motion of the state values and velocities are very low in comparison to the other cost functions of a single term and of two terms, despite the kinetic energy single term that has no range motion on the state values (Figure 11). In terms of the actuators of the three terms cost functions of Figure 14, the trajectories are not smooth and they do not have a great maximum or minimum at 1 second as the other trajectories of the cost functions that include the control goal represented in Figures 12 and 14.

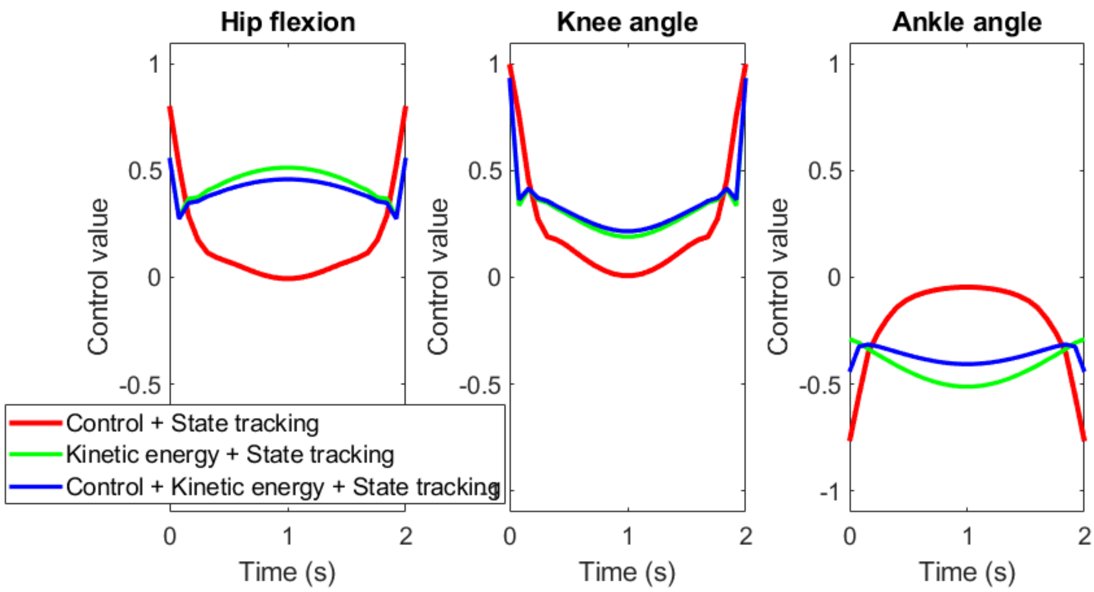
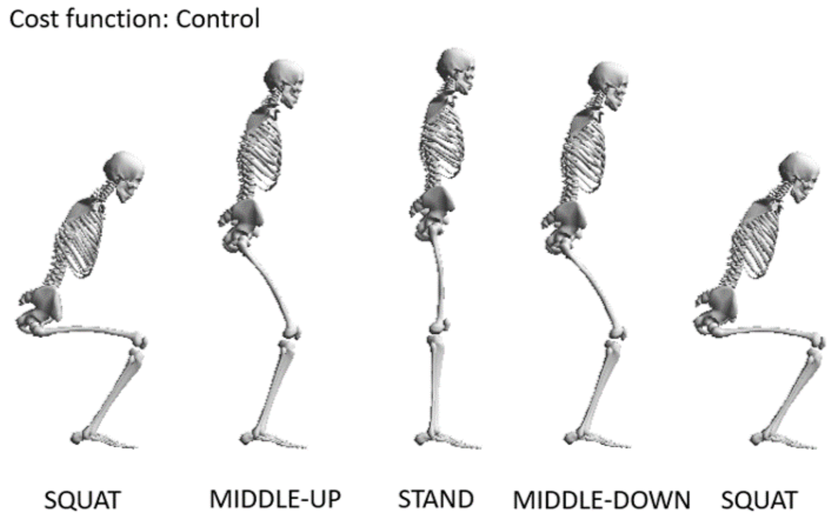
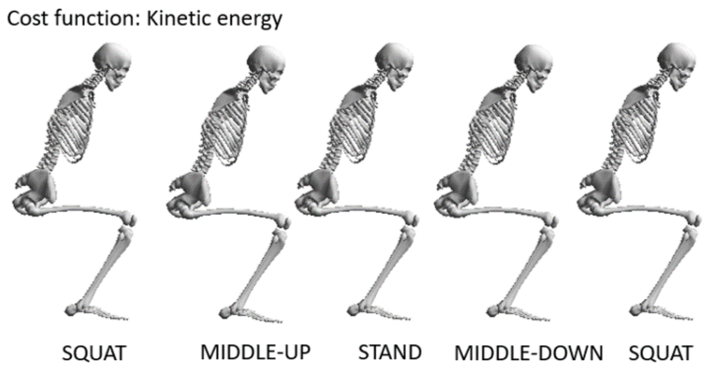


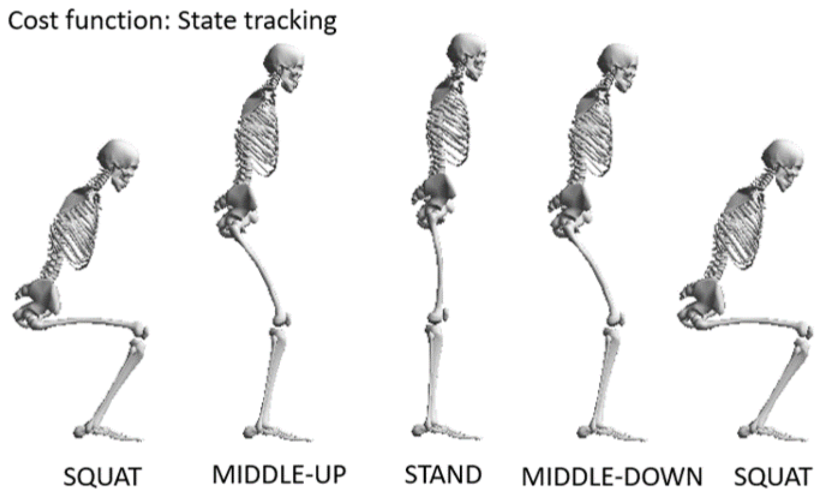

Figure 14: Actuators for every different cost function with two terms. Remember that the y-axis correspond to the control values (from -1 to 1). When this control value is multiplied by the optimal force, the result is the joint torque value.




(a) Images of the squat-to-stand movement using the control cost function (the single term one).




(b) Images of the squat-to-stand movement using the kinetic energy cost function (the single term one).




(c) Images of the squat-to-stand movement using the state tracking cost function (the single term one).

Figure 15: Motion visualization

Despite the use of a few different cost functions, the motion is correctly executed and some valuable conclusions can be reached. Comparing the images of the motion execution (Figures 15a, 15b and 15c), the more correct executed motions are the ones using the single term control function (Figure 15a) and the single term state tracking function (Figure 15c), due to they show the correct motion described at the beginning of the section.

To conclude this section, the obtained results are discussed. On the one hand, as seen in Table 7, the kinetic energy goal has a great contribution to the convergence of the problem, due to the number of iterations decrease with it. On the other hand, the motion with this cost function is directly not executed. As seen in Figure 15b, the model stays at a squat position without changing it. The comment made above of approximately no range motion of the generalized coordinates using the kinetic energy goal confirms the fact that the problem is not well defined at all. Indeed, the states generated with the kinetic energy cost function are constant (Figure 11). For changing this behavior, a kinematic constraint on the model is needed, for instance, the pelvis should arrive at a determine point that matches the standing position. This fact will ensure at least that the movement will be executed. The slight variations on the states between the use of the state tracking and the control term are because in the state tracking solution the states are following approx. the trajectories from the generated data, meanwhile using the control goal, the states are following the values so that the joint torques are minimized. As noticed in Figure 11, the initial and final state values are the same since they were set like this in the formulation problem as variable bounds (Section 5.2). Regarding the cost functions with two and three terms, the cost function with the kinetic energy goal has the ROM of the generalized coordinates much lower than the other cost functions, due to the fact minimizing the kinetic energy cause the minimum movement of the bodies, thus the lower ROMs. The velocities have some maximums and minimums at the beginning of the upward and downward motion using the cost function that does not include the kinetic energy term, thus the kinetic energy is not minimized, thus the velocities take high values at the beginning and in the end of the motion execution. About the controls, the joint torques decrease in absolute value when the kinetic goal is added to the cost function. Thus, the actuator values are lower when the kinetic goal is used. The control variables have a minimum in absolute value at time 1s, in the stand position, because the joint torques are much higher in absolute value in the squat position than in the stand one. To conclude, the best results have been obtained for the cost function of state tracking and control terms, although more research could be done to add some kinetic constraints to the model, thus the kinetic energy cost would have been allowed a correct motion performance. Considering that the data were generated artificially as sinus functions, the results are quite good, despite having real experimental data that will improve the results.

# 5 Walking Prediction

Human gait is the most common and the most studied motion in biomechanics. It is more complex than the squat-to-stand motion. This motion is divided into 5 phases between these six positions: right toe off (RTO), right mid swing (RMS), right heel strike (RHS), left toe off (LTO), left mid swing strike (LMS) and left heel strike (LHS). Every position corresponds to a certain time shown in Table 8 which also contains a description for each position. This motion prediction requires defining the different contact ground forces with constraints or using contact models. As explained in Section 2.1.2, the walking simulations do not correspond to a gait prediction because the 2D model has been simplified. Consequently, it has the pelvis and the torso fixed to the ground. Owing to the model, the movement corresponds to the legs walking while they are "flying". From the whole gait cycle, it has been predicted only the part that corresponds to an 80 % approx. Moreover, in the cost functions, there is always the term of state tracking, due to the motion has been predicted only with the variable bounds and no other restriction. The experimental data used for gait tracking were collected by the research group at the Biomechanics Laboratory (CREB) of *Universitat Politècnica de Catalunya* (UPC) [1].

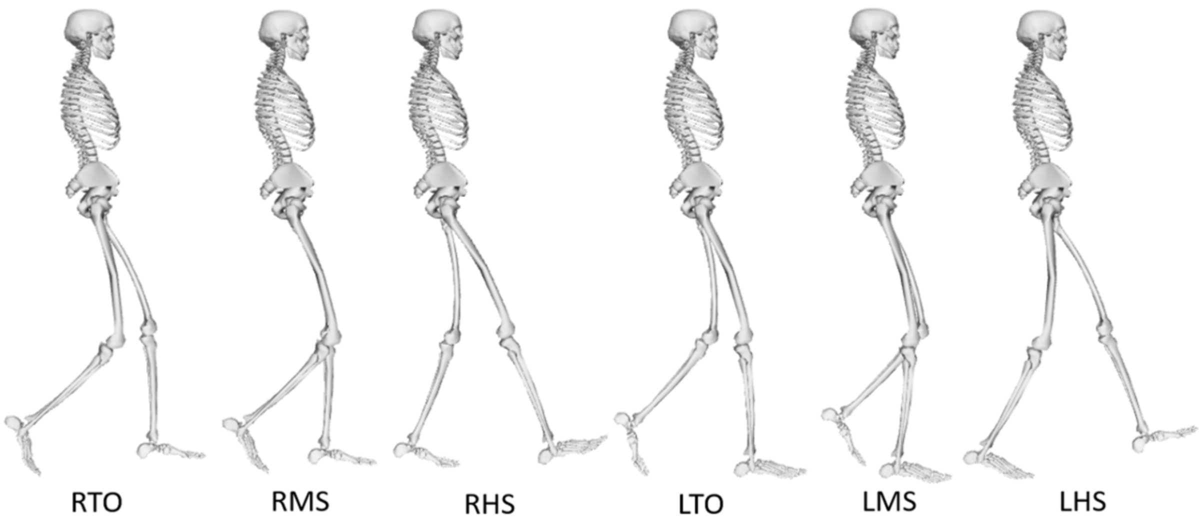



Figure 16: Images of the walking 2D motion using the cost function of state tracking.

| Matching Time (in seconds) | Position's acronym | Position | Description |
|---|---|---|---|
| 0.67 | RTO | Right toe off | Position where the right leg is just initiating the swing |
| 0.75 | RMS | Right mid swing | Position where the right leg is in the middle of the swing |
| 1.07 | RHS | Right heel strike | Position where the right leg is about to put the foot on the floor |
| 1.30 | LTO | Left toe off | Position where the left leg is just initiating the swing |
| 1.42 | LMS | Left mid swing | Position where the left leg is in the middle of the swing |
| 1.75 | LHS | Left heel strike | Position where the left leg is about to put the foot on the floor |

Table 8: Description of the six positions that define the five phases of the walking 2D motion.

### 5.1 Model

The model used is based on another one created by Rajagopal et al. [7] and it was simplified by the research group of the BIOMEC lab. It has the head, arms and trunk modeled as a single body called HAT. It is a 2D model that consists of the HAT, the pelvis and the legs' elements. It has 8 degrees of freedom, 12 rigid bodies and the ground, 12 joints and 8 generalized coordinates. It has the torso and the pelvis fixed to the ground, thus the movement will correspond to the legs walking while they are "flying". Owing to the weld joints of the pelvis and torso with the ground, the joint torque values will not be real.

The 12 rigid bodies are the pelvis, the right and left femurs, the right and left tibias, the right and left talus, the right and left calcn, the right and left toes, and the torso. The model has 12 joints. In Table 9, the joints of the model are list with their name in OpenSim, the degrees of freedom (DOF) that the joint has, the type of joint, the child and parent frames. The model has 8 generalized coordinates which are described in Table 10. Thus, it has 8 degrees of freedom which correspond to the sum of the DOFs of the joints (Table 9).

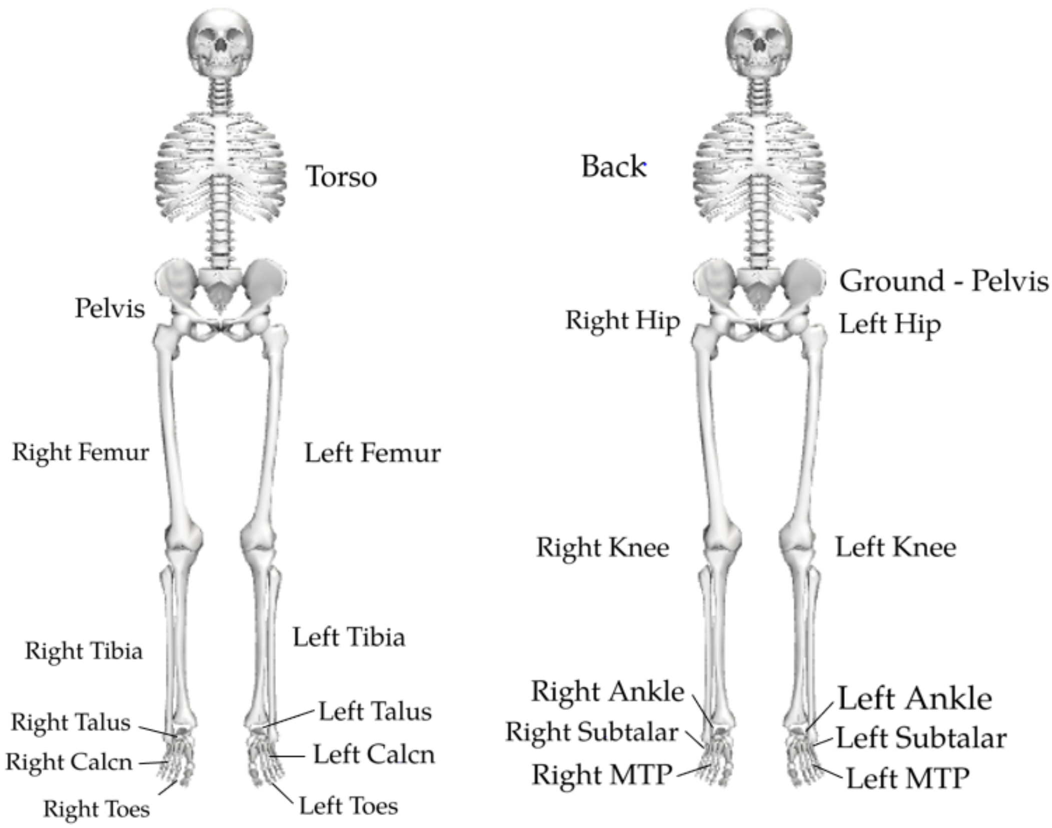



Figure 17: Illustration of the bodies (left) and the joints (right) of the 2D HAT model.

| Joint | Name of the joint in OpenSim | DOF | Type of the joint | Child Frame | Parent Frame |
|---|---|---|---|---|---|
| Ground - Pelvis | $ground\_pelvis$ | 0 | Weld Joint | Pelvis | Ground |
| Right Hip | $hip\_r$ | 1 | Custom Joint | Right Femur | Pelvis |
| Right Knee | $walker\_knee\_r$ | 1 | Custom Joint | Right Tibia | Right Femur |
| Right Ankle | $ankle\_r$ | 1 | Pin Joint | Right Talus | Right Tibia |
| Right Subtalar | $subtalar\_r$ | 0 | Weld Joint | Right Calcn | Right Talus |
| Right MTP | $mtp\_r$ | 1 | Pin Joint | Right Toes | Right Calcn |
| Left Hip | $hip\_l$ | 1 | Custom Joint | Left Femur | Pelvis |
| Left Knee | $walker\_knee\_l$ | 1 | Custom Joint | Left Tibia | Left Femur |
| Left Ankle | $ankle\_l$ | 1 | Pin Joint | Left Talus | Left Tibia |
| Left Subtalar | $subtalar\_l$ | 0 | Weld Joint | Left Calcn | Left Talus |
| Left MTP | $mtp\_l$ | 1 | Pin Joint | Left Toes | Left Calcn |
| Back | $back$ | 0 | Weld Joint | Torso | Pelvis |

Table 9: 2D HAT model joints, their name in OpenSim, the degrees of freedom of the child frame with respect to the father frame and the type of the joint. See the joint location on the model in Figure 17.

| Name of the generalized coordinate in OpenSim | Matching Joint | Description |
|---|---|---|
| $hip\_flexion\_r$ | Right Hip | The angle that has the right femur with respect to the pelvis |
| $knee\_angle\_r$ | Right Knee | The angle that has the right tibia with respect to the right femur |
| $ankle\_angle\_r$ | Right Ankle | The angle that has the right talus with respect to the right tibia |
| $mtp\_angle\_r$ | Right MTP | The angle that has the right toes with respect to the right calcn |
| $hip\_flexion\_l$ | Left Hip | The angle that has the left femur with respect to the pelvis |
| $knee\_angle\_l$ | Left Knee | The angle that has the left tibia with respect to the left femur |
| $ankle\_angle\_l$ | Left Ankle | The angle that has the left talus with respect to the left tibia |
| $mtp\_angle\_l$ | Left MTP | The angle that has the left toes with respect to the left calcn |

Table 10: The name of the generalized coordinates of the 2D HAT model in OpenSim nomenclature, their matching joint and their description.

### 5.2 Problem Formulation

In this section, the state and control variables are presented. Furthermore, the different cost functions and constraints are defined.

There are 16 states: 8 generalized coordinates and 8 velocities of the following angles: the right and left hip flexions, the right and left knee angles, the right and left ankle angles, and the right and left MTP angles.

There are 8 control variables. Table 11 shows the name in Opensim of the actuators, their corresponding coordinate name in OpenSim of the model and the optimal force that is established to the actuator. The minimum and maximum control value of all the actuators of Table 11 is set to -1 and +1. In summary, 8 actuators are added which corresponds to the 8 DOFs of the model.

| Actuator | Matching Coordinate | Optimal Force [in N] |
|---|---|---|
| $tau\_hip\_flexion\_r$ | Right Hip | 120 |
| $tau\_knee\_angle\_r$ | Right Knee | 30 |
| $tau\_ankle\_angle\_r$ | Right Ankle | 10 |
| $tau\_mtp\_angle\_r$ | Right MTP | 10 |
| $tau\_hip\_flexion\_l$ | Left Hip | 120 |
| $tau\_knee\_angle\_l$ | Left Knee | 30 |
| $tau\_ankle\_angle\_l$ | Left Ankle | 10 |
| $tau\_mtp\_angle\_l$ | Left MTP | 10 |

Table 11: Description of the actuators, with their matching coordinates and their optimal force.

The initial and final time bounds are set to 0.67 and 1.75 seconds, which are the times when the first and final value of the experimental data was taken.

Then, the cost terms (explained in Section 3.4) are added to the problem. Three different cost functions are applied and studied in the walking 2D problem: state tracking, or state tracking with control or kinetic energy. The control term minimizes the effort of the controls, that is the effort of the joint torques of the 8 actuators. The kinetic energy term minimizes the kinetic energy that the model requires to execute the motion. And the state tracking term minimizes the squared difference between the predicted and experimental states (coordinates and velocities). For more information about this data see Section 3.3. Furthermore, there is an example in the Moco Documentation that does tracking differently, that example's concepts are based on the article [25].

The constraints of this problem are the variable bounds. The position bounds of the generalized coordinates are the real bounds of the joints determined by OpenSim, whereas the initial and final bounds are the same as the experimental data. All these values are shown in Table 12. The bounds of the velocities of the generalized coordinates are shown in Table 13 and they have been set with following this equation:

$$[min, max] = [10 \times (min(\dot{q}_i) - 0.5 \times range), 10 \times (max(\dot{q}_i) + 0.5 \times range)]$$

where $range = |max(\dot{q}_i) - min(\dot{q}_i)|$ and $\dot{q}_i$ is the velocity of the generalized coordinate $i$.

| Name of the generalized coordinate in OpenSim | Trajectory bounds [deg] | Initial bound [deg] | Final bound [deg] |
|---|---|---|---|
| $hip\_flexion\_r$ | [-30, 120] | 4.97 | -5.94 |
| $hip\_flexion\_l$ | [-30, 120] | 20.54 | 22.26 |
| $knee\_angle\_r$ | [-0.57, 120] | 50.68 | 20.99 |
| $knee\_angle\_l$ | [-0.57, 120] | 16.88 | -0.04 |
| $ankle\_angle\_r$ | [-40, 30] | -18.03 | 14.94 |
| $ankle\_angle\_l$ | [-40, 30] | -2.22 | -2.38 |
| $mtp\_angle\_r$ | [-30, 30] | -15.35 | -3.75 |
| $mtp\_angle\_l$ | [-30, 30] | -3.94 | -9.25 |

Table 12: Position bounds of the generalized coordinates of the walking 2D problem.

| Name of the generalized coordinate in OpenSim | Trajectory bounds [deg/s] | Initial bound [deg/s] | Final bound [deg/s] |
|---|---|---|---|
| $hip\_flexion\_r$ | [-2842.73, 2821.63] | 127.66 | -146.96 |
| $hip\_flexion\_l$ | [-2477.74, 3082.35] | -76.17 | -104.05 |
| $knee\_angle\_r$ | [-6699.13, 4808.63] | 193.17 | -12.94 |
| $knee\_angle\_l$ | [-7518.93, 5478.34] | -14.89 | -4.66 |
| $ankle\_angle\_r$ | [-3692.11, 3715.88] | 54.96 | -22.82 |
| $ankle\_angle\_l$ | [-5210.20, 4754.61] | 60.25 | -30.66 |
| $mtp\_angle\_r$ | [-2215.74, 3917.65] | 238.43 | -23.72 |
| $mtp\_angle\_l$ | [-4090.28, 4453.25] | 71.77 | 19.47 |

Table 13: Velocity bounds of the generalized coordinates of the walking 2D problem.

The initial guess used is the Moco's default one in which each variable's value is the midpoint of the variable's bounds. The number of mesh intervals is 25, and the convergent and constraint tolerances are set to $10^{-4}$.

### 5.3 Results and discussion

| Cost function | | | Convergence | | Cost function values | | | |
|---|---|---|---|---|---|---|---|---|
| Control | KE | StateTrack | Num. iter. | Time [s] | Cost function | Control | KE | StateTrack |
| | | X | 17 | 24 | 0.11 | - | - | 0.11 |
| X | | X | 15 | 37 | 0.56 | 0.32 | - | 0.24 |
| | X | X | 16 | 43 | 2.83 | - | 1.72 | 1.11 |

Table 14: Table of different aspects of the results for the different cost functions studied in the walking 2D motion.

Tables 14 and 15 illustrate the results of the prediction of walking 2D motion. In terms of convergence, in Table 14, all the cost functions have approximately the same number of iterations, but it should be said that the state tracking cost function has the lowest duration of the simulation (24s). This function takes less effort than the other ones. There is a cost function that stands out from the others and that corresponds to the one of state tracking and kinetic energy with an effort of 2.83.

| Cost function | | | ROM [deg] | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Control | KE | StateTrack | Right Hip | Left Hip | Right Knee | Left Knee | Right Ankle | Left Ankle | Right MTP | Left MTP |
| | | X | 30.29 | 33.08 | 59.23 | 57.5 | 33.16 | 33.24 | 20.68 | 30.96 |
| X | | X | 30.07 | 32.15 | 59.19 | 57.26 | 33.16 | 33.24 | 20.67 | 30.91 |
| | X | X | 20.23 | 19.97 | 54.22 | 51.49 | 33.16 | 32.96 | 20.67 | 30.92 |

Table 15: Table of different aspects of the ranges of motion of the state values for the different cost functions studied in the walking 2D motion.

In Table 15, the ranges of motion (ROM) are illustrated for each generalized coordinate. It is appreciated that there is no difference in the ROM between the cost function of state tracking as a single term and state tracking with control (with an absolute error from 0 to 0.93 deg). However, using the cost function of state tracking with the kinetic energy goal, the ROM of the right and left hips vary a little, with an absolute error of 0.93 deg. That fact is seen also in Figure 18, where it is illustrated that there is no variation between the trajectories of the 16 states, despite the right and left hip flexions that have an slight variation.

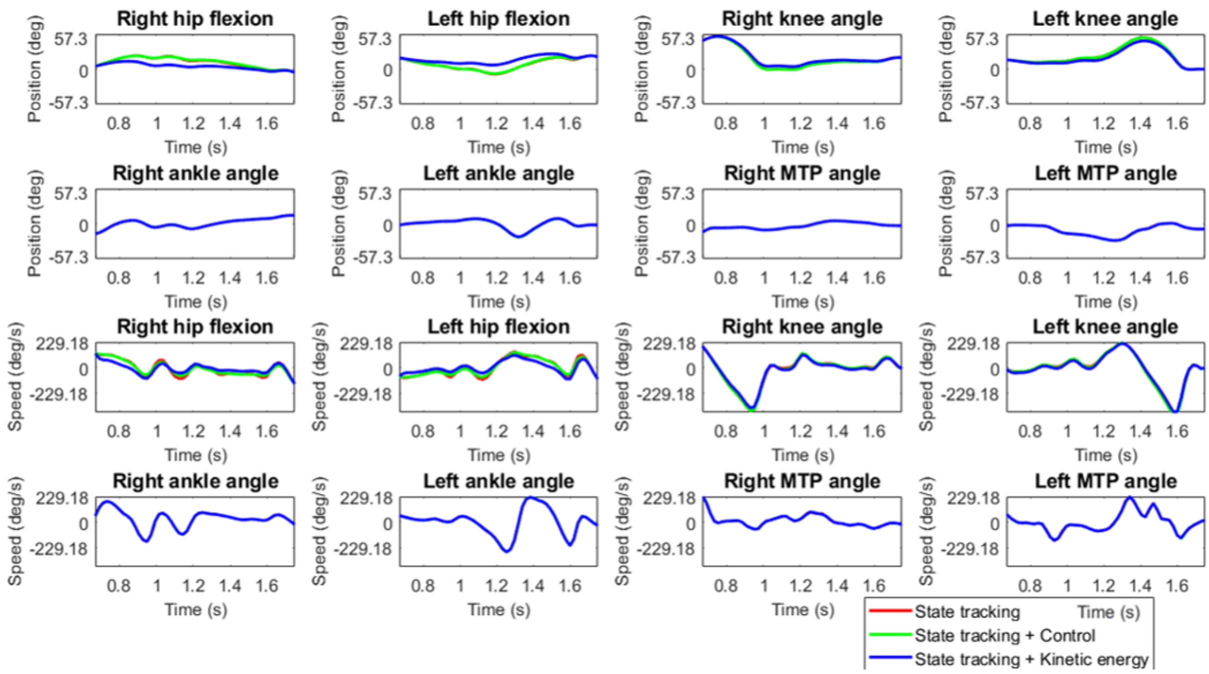



Figure 18: States (generalized coordinates and velocities) for every different cost function.

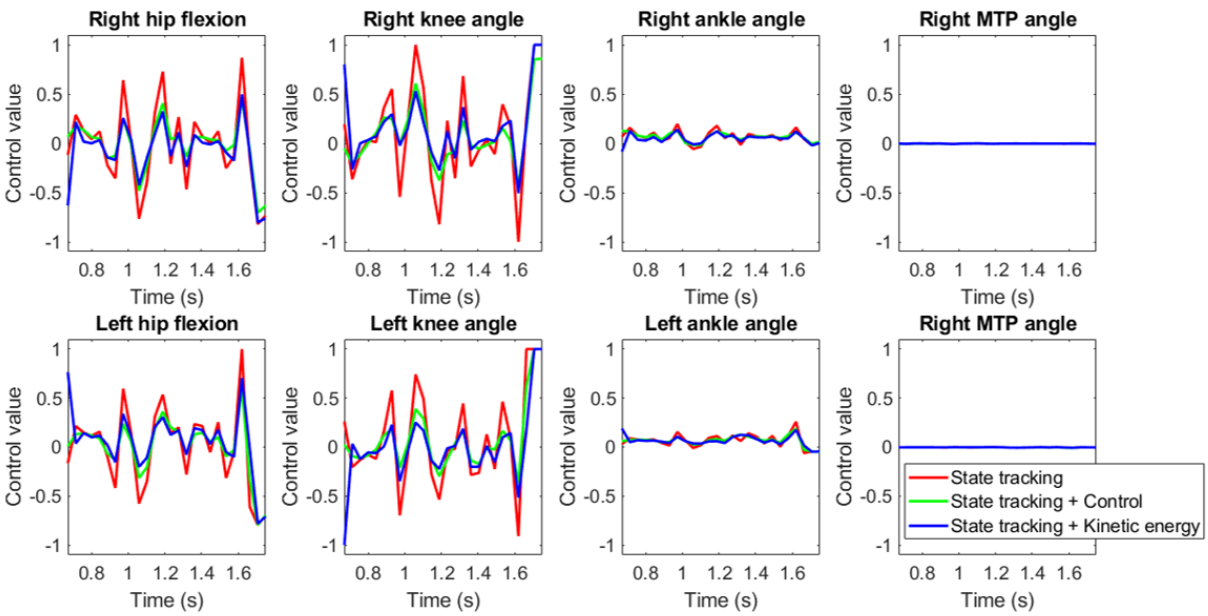
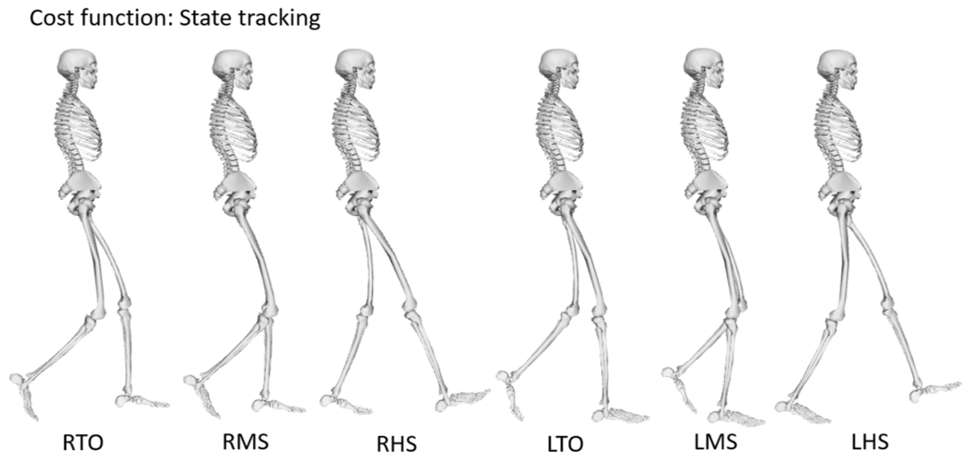


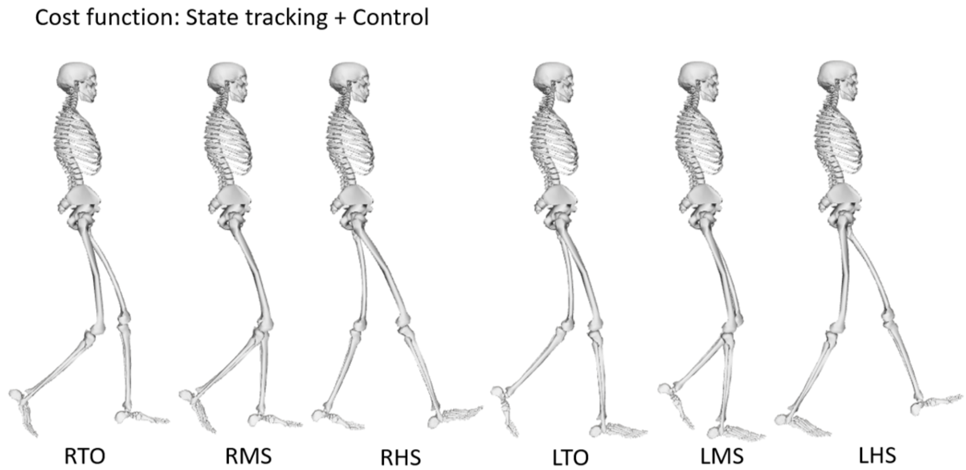

Figure 19: Actuators for every different cost function. Remember that the y-axis correspond to the control values (from -1 to 1). When this control value is multiplied by the optimal force, the result is the joint torque value.

In Figure 19, it can be appreciated the trajectories of the joint torques that have the role of actuators on the walking 2D problem. Although the trajectories' shapes for the different cost functions are similar, Figure 19 shows that there is a variation regarding the different cost functions. It is a significant variation because the value represented in Figure 19 refers to the control value (from -1 to 1) multiplied by the optimal force of the actuator. It stands out that the joint torques of the right and left MTPs, are approximately zero. For the state tracking cos function, the maximum values for the joint torques are approx. 84Nm for the right hip, 120Nm for the left hip, 30Nm for the right and left knees, 2Nm for the right and left ankles and approx. 0Nm for the right and left MTPs.

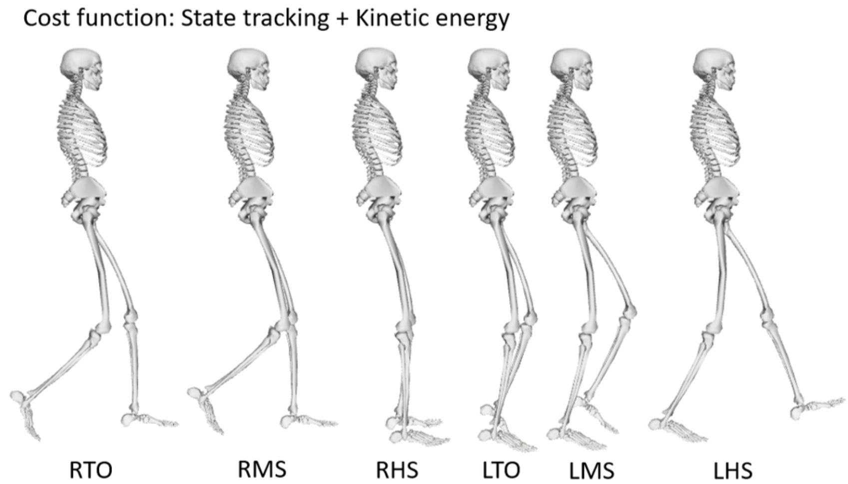
In terms of motion execution, it is seen in Figures 20a and 20b that the motion is correctly executed and every position of the walking 2D motion corresponds to the image above the title. For instance, the "LTO" image shows the model taking off the left foot from the floor. Moreover, it is the same position as the "RTO" image but with the other foot. Figure 20c illustrates the images of the motion using state tracking with the kinetic energy. It is appreciated that the right foot does not execute a full step, concretely in the "RHS" position the right leg should be extended to the front of the model about to touch the floor with the right foot. Minimizing the kinetic energy, in this case, does not bring to the right execution of the motion, but not as wrong as in the squat-to-stand problem.




(a) Images of the walking 2D motion using the cost function of state tracking.




(b) Images of the walking 2D motion using the cost function that combines state tracking and control.




(c) Images of the walking 2D motion using the cost function that combines state tracking and kinetic energy.

Figure 20: Motion visualization

To conclude this section the obtained results are discussed. Adding a cost term to the state tracking goal does not improve convergence in terms of the number of iterations, but it is a bit worse in terms of time duration indeed. It makes sense that the effort value of the state tracking cost function is much lower than the combination of it with another goal because other magnitudes have to be analyzed, consequently, this evokes more effort. However, increasing the time of simulations is not an issue if the motion prediction is better. In fact, for motion prediction, it is interesting to remove the state tracking term or simply establishing a small weight. In addition, the state tracking combined with the kinetic energy goal has a high cost function effort, because following a reference trajectory meanwhile the kinetic energy is being minimized takes a lot of effort comparing to only do state tracking. Regarding motion execution, it can be said that adding the control goal to the state tracking one does not improve performance because they make a similar motion. Furthermore, notice that in Figure 20a the "LHS" image has the left foot a bit higher than the right foot on the "RHS" position, that is because the experimental data is taken of an 80% of the full cycle. The variation of the ROM of the right and left hips of the state tracking cost function with the kinetic energy goal explains the incorrect execution of the gait, whereas the low difference between the ROMs of the state tracking cost function with the one combined with the control term means that the control goal does not bring information at all. About the controls, it is normal that the MTPs torques remain constant since these joint does not have to take part in the motion execution. Moreover, the hip and the knee torques change over time because they are key elements for the gait. To conclude, the state tracking is the best way to predict gait motion and the other goals do not bring any extra information. However, in order to predict the complete gait cycle, control and kinetic energy goals should be implemented with some kinematic constraints, using a full-body model and a contact model. For completely motion prediction, state tracking goal should not be used, and if it is used its weight should be very low.

# 6 Project Impact

Environmental impact

The environmental impact has been much lower than planned at the beginning because it has not been used all the equipment to obtain the experimental data. Despite that, a laptop has been used with a lot of electric energy consumption.

Social impact

This thesis does not have a direct social impact. However, future research in the line of this project will have it. For instance, implementing the lab equipment for obtaining data from patients in a hospital and predict their walking motion, acquiring a lot of data of internal biomechanic values. Moreover, some scientists that have already applied control techniques to biomedicine, as Swan already exposed in his book [23].

Economic impact

The economic cost of the project is the sum of the personal laptop, MATLAB and MOCO licenses, electric energy and working time of the students and supervisors.
Regarding the laptop, it is estimated 5 years of useful life and a price of 700€. It is considered to be used during 52 weeks per year, 10 hours per day and 5 days per week, which is in total 13000 hours of useful life. It results in a variable cost of 0.0538€/h. About the Moco license is completely free, meanwhile, the MATLAB license cost is 250€ for academic use and it annually expires. Having this license 52 weeks per year, 7 days per week and 24 hours a day, results in 1456 hours of useful life, which end up to a variable cost of 0.1717€/h.
In terms of working hours, the student has dedicated 250 hours in total (earning 8€/h), which corresponds also to the hours that the laptop has been consuming electric energy, because even the meetings were by video calls. The supervisors' hours are estimated at 30 hours and earned 50€/h.
As for the electric energy, only the electric energy consumed by the laptop has to be taken into account. The power of the laptop is 65W and it has been estimated a constant value of electric energy equal to 0.12€/kWh. Thus, the variable cost of electric energy is 0.0078€/h.
The total cost of the project is 4451€, see Table 16.

| Cost factor | Useful life or work time [h] | Cost per hour [€/h] | Cost related to the project |
|---|---|---|---|
| Laptop | 13000 | 0.0538 | 699.40 |
| MATLAB license | 1456 | 0.1717 | 250.00 |
| Student | 250 | 8 | 2000.00 |
| Supervisors | 30 | 50 | 1500 |
| Electric energy | 210 | 0.0078 | 1.64 |
| | | **TOTAL COST** | **4451.04** |

Table 16: Calculation of the final cost of the project.

## Conclusions

To conclude this thesis, the objectives have been reached. The acquiring of knowledge have been done and the codes have been successfully codified, thus the squat-to-stand and the walking 2D motion have been predicted. Results are valued positively and they lead to an interesting discussion, especially between the different cost functions used. The kinetic energy goal is the best cost term for improving convergence. However, the best cost function for correct motion execution is state tracking. Furthermore, the squat-to-stand motion has been correctly executed and it corresponds to real motion. In contrary, the gait predicted in this thesis is not a real motion due to nor a contact model neither a full-body model has not been used. That is the reason why the joint torque values are not real. Adding the control goal does not bring any advantage in front of the state tracking term, but it will be useful for real motion prediction when the state tracking is removed. In short, although tracking causes the most correct motion execution, motion prediction should be done without it.

As far as I am concerned, I have learned to make a research about a field which was quite new for me at the beginning. Once I got used to the target vocabulary of optimal control for human motion prediction, I had to learn to code with MATLAB using the Moco software which was the more difficult task for me in this project. Consequently, I persevered improving my knowledge about the field and finally having the results I obtained. I have learned to be constant at working at home without the pressure of nobody, to keep searching when I do not understand something instead of throwing it all away and also to be patient if the code does not run properly, if the simulation does not converge or if I do not obtain the results I want. Despite all the inconvenient issues I had, to be honest it has been worth doing this work, especially because it makes me feel fulfilled putting a grain of sand to the research of the human motion prediction. I am sure that in the future it will help to improve people's health.

I have had some complications during my thesis. Due to the pandemic of COVID-19, I have had to work from home without having the opportunity to tell the thesis issues to my tutors face to face, instead, through a video call and by mail. Moreover, instead of a more efficient computer of the lab, I must work with my laptop which I had to clean up from files in order MATLAB can process better for making simulations. These issues had made me start to work on my thesis later than I have planned at the beginning. Furthermore, I could not get any experimental data since the CREB lab was closed, as well as the UPC university.

In order to progress on the line of this thesis, some improvements can be implemented in the future. On the one hand, simulating the walking 2D motion with a full-body model including arms and later with a 3D model to predict a 3D gait motion. The full-body model includes the kinematic restrictions and the contact model that allows the feet-ground contact. Furthermore, different formulations can be implemented, for instance, defining the torque joints and the accelerations as states and the jerks (the third derivative of the generalized coordinates) as control variables. Moreover, to predict gait with a model with muscles could improve the prediction. And, on the other hand, knowing the C++ language could help to learn how to use Moco functions and C++ brings more possibilities than MATLAB. Finally, another improvement is to prototype your own custom goal for establishing the cost function as the research question demands.

## Acknowledgments

First of all, my sincere gratitude towards my supervisors, Josep Maria and Míriam. Thank you for their advice and their dedication, especially the rapid answers to my mails. Thank you too for trusting me to do this project.

In addition, I would like to thank the BIOMEC lab of the *Universitat Politècnica de Catalunya* and the research group, especially thank you for their sympathy in the first days of work and to make to be part of this project easy.

## References

[1] FEBRER NAFRÍA, M. *Prediction of assisted walking using subject-specific biomechanical models.* Research plan of PhD Thesis, Biomedical Engineering Doctoral Programme, Universitat Politècnica de Catalunya. 2017.

[2] PALLARÈS LÓPEZ, R. *Optimal Control Prediction of Dynamic Consistent Walking Motions*, bachelor's thesis, Universitat Politècnica de Catalunya (UPC), Spain, June 2017.

[3] SETH, A., HICKS, J.L., UCHIDA, T.K., HABIB, A., DEMBIA, C.L., DUNNE, J.J., ONG, C.F., DEMERS, M.S., RAJAGOPAL, A., MILLARD, M., HAMNER, S.R., ARNOLD, E.M., YONG, J.R., LAKSHMIKANTH, S.K., SHERMAN, M.A., KU, J.P. AND DELP, S.L. *OpenSim: Simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement*, Hebrew University of Jerusalem, Israel, July 2018. *PLoS Computational Biology* 14(7): e1006223. DOI: 10.1371/journal.pcbi.1006223.

[4] DELP, S.L., ANDERSON, F.C., ARNOLD, A.S., LOAN, P., HABIB, A., JOHN, C.T.,GUENDELMAN, E. AND THELEN, D.G. *OpenSim: Open-Source Software to Create and Analyze Dynamic Simulations of Movement*, IEEE Transactions on Biomedical Engineering, VOL. 54, NUM. 11, November 2007.

[5] DEMBIA, C.L., BIANCO, N.A., FALISSE, A., HICKS, J.L. AND DELP, S.L. *OpenSim Moco: Musculoskeletal optimal control*, Cold Spring Harbor Laboratory, November 2019. DOI: 10.1101/839381.

[6] OpenSim Moco Documentation. Available at: <https://opensim-org.github.io/opensim-moco-site/docs/0.3.0/index.html> [Online; accessed 21 April 2020]

[7] RAJAGOPAL, A., DEMBIA, C.L., DEMERS, M.S., DELP, D.D., HICKS, J.L. AND DELP, S.L. *Full-Body Musculoskeletal Model for Muscle-Driven Simulation of Human Gait*. IEEE Transactions on Biomedical Engineering, VOL. 63, NUM.10, PAGES 2068-2079, October 2016.

[8] OpenSim API 4.0. Available at: <https://simtk.org/api_docs/opensim/api_docs40 /index.html> [Online; accessed 25 April 2020]

[9] TANG, Y. AND HUI, K-C. *The effect of tendons on foot skin deformation.* Computer-Aided Design, July 2007. DOI: 10.1016/j.cad.2007.01.013.

[10] ANDERSSON, J. A. E., GILLIS, J., HORN, G., RAWLINGS, J. B. AND DIEHL, M. *CasADi – A software framework for nonlinear optimization and optimal control.* Springer Verlag. Mathematical Programming Computation. 2019. VOL. 11, ISS. 1, PAGES 1-36. DOI: 10.1007/s12532-018-0139-4.

[11] WÄCHTER, A. AND BIEGLER, L. T. *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming.* Springer. Mathematical programming. 2006. VOL. 106, PAGES 25-57. DOI: 10.1007/s10107-004-0559-y.

[12] GILL, P. E., MURRAY, W. AND SAUNDERS, M. A. *SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization.* SIAM Review. 2005. 47(1), PAGES 99-131. DOI: 10.1137/s0036144504446096.

[13] KELLY, M. *An Introduction to Trajectory Optimization: How to Do Your Own Direct Collocation.* SIAM Review. 2017. VOL. 59, NUM. 4, PAGES 849-904. DOI: 10.1137/16M1062569.

[14] DORN, T.W., SCHACHE, A.G. AND PANDY, M.G. *Muscular strategy shift in human running: dependence of running speed on hip and ankle muscle performance.* 2012. J Exp Biol 2015, 1944-56. DOI: 10.1242/jeb.064527.

[15] DORN, T.W., WANG, J.M., HICKS, J.L. AND DELP, S.L. *Predictive Simulation Generates Human Adaptations during Loaded and Inclined Walking.* Journal Plos One, 2015. DOI: 10.1371/journal.pone.0121407.

[16] STEELE, K.M., VAN DER KROGT, M.M., SCHWARTZ, M.H. AND DELP, S.L. *How Much Muscle Strength Is Required to Walk in a Crouch Gait?* 2012. Journal of biomechanics 45(15):2564-9. DOI: 10.1016/j.jbiomech.2012.07.028.

[17] PRIAMIKOV, A., FRONIUS, M., SHI, B. AND TRIESCH, J. *OpenEyeSim: A biomechanical model for simulation of closed-loop visual perception.* 2016. Journal of Vision 16(15):25. DOI: 10.1167/16.15.25.

[18] CASPI, R., BILLINGTON, R., KESELER, I.M., KOTHARI, A., KRUMMENACKER, M., MIDFORD, P.E., ONG, W.K., PALEY, S., SUBHARAVETI, P. AND KARP, P.D. *The MetaCyc Database of Metabolic Pathways and Enzymes - A 2019 Update.* 2020. Nucleic acids research, vol. 48, D1, D445-D453. DOI: 10.1093/nar/gkz862.

[19] UMBERGER, B.R. AND MILLER, R.H. *Optimal Control Modeling of Human Movement.* From *Handbook of Human Motion.* Springer, 2018. DOI: 10.1007/978-3-319-30808-1_177-1.

[20] ONG, C., GEIJTENBEEK, T., HICKS, J., DELP, S. *Predicting gait adaptations due to ankle plantarflexor muscle weakness and contracture using physics-based musculoskeletal simulations.* 2019. DOI: 10.1101/597294.

[21] ACKERMANN, M. AND VAN DER BOGERT, A.J. *Optimality principles for model-based prediction of human gait.* 2010. Journal of Biomechanics, 43(6):1055-1060. DOI: 10.1016/j.jbiomech.2009.12.012.

[22] D'ANDREA, R., 2019. *Dynamic Programming and Optimal Control.* Lecture Notes. ETH, Zürich. Unpublished.

[23] SWAN, G.W. *Applications of optimal control theory in biomedicine.*, 1984. New York, M. Dekker.

[24] EZATI, M., GHANNADI, B. AND MCPHEE, J. *A review of simulation methods for human movement dynamics with emphasis on gait.* 2019. Multibody System Dynamics, 47, 265-292. DOI: 10.1007/s11044-019-09685-1.

[25] FALISSE, A., SERRANCOLÍ, G., DEMBIA, C.L., GILIS, J. AND DE GROOTE, F. *Algorithmic differentiation improves the computational efficiency of OpenSim-based trajectory optimization of human movement.* 2019. PLoS ONE 14(10): e0217730. DOI: 10.1371/journal.pone.0217730.

[26] NORMAN-GERUM, V.T. *Predictive Dynamic Simulation of Healthy Sit-to-Stand Movement.* Thesis, University of Waterloo, Ontario, Canada, 2019.

[27] FREGLY, B.J., FREEGLY, C.D. AND KIM, B.T. *Computational Prediction of Muscle Moments During ARED Squat Exercise on the International Space Station.* 2015. Journal of Biomechanical Engineering, 137:121005. DOI: 10.1115/1.4031795.

[28] SCHELLENBERG, F., TAYLOR, W., TREPCZYNSKI, A., LIST, R., KUTZNER, I, SCHÜTZ, P., DUDA, G. AND LORENZETTI, S. *Evaluation of the accuracy of musculoskeletal simulation during squats by means of instrumented knee prostheses.* 2018. Medical Engineering Physics, 61. DOI: 10.1016/j.medengphy.2018.09.004.

[29] MEYER, A.J., ESKINAZI, I., JACKSON, J.N., RAO, A.V., PATTEN, C. AND FREGLY, B.J. *Muscle Synergies Facilitate Computational Prediction of Subject-Specific Walking Motions.* 2016. Frontiers in Bioengineering and Biotechnology, VOL. 4. DOI: 10.3389/fbioe.2016.00077.

[30] SCHOENFELD, B. *Squatting Kinematics and Kinetics and Their Application to Exercise Performance.* 2010. Journal of Strength and Conditioning Research, VOL. 24, ISSUE 12, PAGES 3497-3506. DOI: 10.1519/JSC.0b013e3181bac2d7.

[31] CASTERMANS, T., DUVINAGE, M., CHERON, G. AND THIERRY, D. *Towards Effective Non-Invasive Brain-Computer Interfaces Dedicated to Gait.* 2013. Rehabilitation Systems. Brain Sciences, 4, 1-48. DOI: 10.3390/brainsci4010001.

[32] OTTEN, E. *Inverse and forward dynamics: Models of multi-body systems.* 2003. Philosophical transactions of the Royal Society of London. Series B, Biological sciences, 358, 1493-500. DOI: 10.1098/rstb.2003.1354.

[33] Moco Examples. Available at: <https://opensim-org.github.io/opensim-moco-site/docs/> [Online; accessed 21 June 2020]